JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 27, No. 3, May–June 2004

Optimal State Estimation with Failed Sensor Discrimination and Identification

Michael E. Polites,* Kevin E. Witzberger,† Christopher M. Lane,† and Mark N. Thornblom†
University of Alabama, Tuscaloosa, Alabama 35487

A new estimation scheme is presented that combines a fixed-gain Kalman filter for optimal state estimation with a prefilter that discriminates against failed sensors and identifies a failed sensor in real time. This new scheme has features characteristic of systems with triple-redundant sensing and voting, but with fewer sensors. It is tested on second- and third-order plants with dual-redundant measurements of the system states and is shown to out perform the stand-alone Kalman filter by a factor of two or more in terms of the rms estimation errors. Strategies for application to systems higher than third order are discussed.

I. Introduction

KALMAN filters are widely used for optimal state estimation.[1] They can be found in spacecraft, aircraft, missiles, and weapon systems, among others. However, for the Kalman filter to generate good estimates of the system states, the sensors, which generate the measurements that are input into the Kalman filter, must be working properly. Otherwise, the estimated states will be in error. To insure that measurements from failed sensors are not used in the estimation process, there is typically a separate software routine that operates in parallel with the Kalman filter, or with whatever other state estimator is being used, and that makes checks and comparisons of the sensor outputs to try to determine if a sensor has failed and, if so, which one. If a sensor fails and the failure is detected, this software routine then issues commands to isolate the failed sensor from the estimation process and substitute another one in its place, or to reconfigure the estimation software to operate without the failed sensor. These software routines are sometimes called redundancy management systems, failure detection and isolation systems, or failure detection and reconfiguration systems.[2–4] The problem with this approach is that detection of a failed sensor and reconfiguration of the sensors or software must happen instantaneous to prevent the estimation process from failing, and this is impossible. A technique commonly used with spacecraft is to put the spacecraft into some type of safe mode when a failure is detected and wait for help from the ground. Consider the Hubble Space Telescope[5] and the Chandra Advanced X-Ray Astrophysics Facility.[6] These are two NASA free-flying spacecraft that are presently in Earth orbit and performing well. Each spacecraft has a software routine in its onboard flight computer for sensor redundancy management. If a problem is detected with a sensor, the redundancy management system issues commands that put the spacecraft into one of several possible safe modes, depending on the severity of the anomaly. For less critical anamolies, the spacecraft simply enters an attitude hold mode and waits for help from the ground. If the problem is more severe, the spacecraft slews to a sun point attitude for maximum electrical power and maintains that attitude until the ground can assist. If the problem is very severe, backup sensors and an alternate flight computer with unique software are used to slew the spacecraft to the sun point attitude and hold it there until the ground can intervene. Ground intervention involves ground controllers analyzing telemetry data to confirm that a sensor has failed and to determine which one. They then uplink commands that isolate the failed sensor from the estimation process and bring another sensor online or reconfigure the onboard estimation software for the new sensor configuration. Finally, they uplink commands to restore the spacecraft to normal operation. Such a process takes a long time to execute and complete, typically on the order of 12 h.

As one might suspect, there are places where this amount of time for sensor failure detection and isolation is simply intolerable. Consider a launch vehicle with astronauts onboard as in the space shuttle or the Saturn V. During ascent, the system states need to be estimated correctly, in spite of any sensor failures, and sensor failure detection and isolation needs to be instantaneous and infallible. In this case, the usual approach is to employ triple redundant sensing and a scheme called voting.[7] Here, the outputs of three sensors of a given type are compared. Should all outputs match well, the average of the three is input into the estimator. If two match and a third does not, it is concluded that the sensors with matching outputs are working properly, whereas the third is not. Then, the average of the first two outputs is input into the estimator and the third one is ignored. This approach discriminates against failed sensors and offers real-time failure detection, but is very expensive to implement because it requires a lot of sensors.

Another problem with a Kalman filter is that it takes some time for the estimation errors to reach their minimum steady-state values, especially when there is a good deal of uncertainty in the initial state of the system. This is true even when the standard time-varying Kalman filter gains are used. It is especially true when the fixed steady-state Kalman filter gains are employed.

Yet another problem with a Kalman filter is that sometimes sensor measurements are generated faster than the Kalman filter algorithms can process them. Then, some of the measurements must be discarded, which is wasteful. To avoid wasted measurements, calculation of their average in a prefilter has been proposed.[8] The idea is to average available measurements over the time interval between Kalman filter computational cycles. This reduces measurement noise, but also averages the true measurements, creating distorted measurements that are input into the Kalman filter. The result is an additional component of error in the estimated states of the system.

Thus, three problems associated with Kalman filters include the following:

1) The sensors, which generate measurements that are input into the Kalman filter, must work properly, and schemes for detecting and isolating failed sensors work slower than necessary unless triple-redundant sensing is used.

2) It takes some finite time for the estimation errors to reach a minimum.

3) Sometimes, sensor measurements are generated faster than the Kalman filter can process them, and some must be discarded, which is wasteful.

Received 31 March 2003; revision received 7 August 2003; accepted for publication 25 October 2003.

*Associate Professor, Department of Aerospace Engineering and Mechanics. Associate Fellow AIAA.

†Research Assistant, Department of Aerospace Engineering and Mechanics. Member AIAA.

This paper proposes a Kalman filtering scheme with none of these problems and features the benefits of triple-redundant sensing and voting with fewer sensors. It works as follows: An undistorted estimate of the system state is generated for every computation cycle of the Kalman filter, with a prefilter that processes all of the sensor measurements generated over each computation cycle of the Kalman filter. Hence, none of the sensor measurements are wasted. A key element of the prefilter is a state reconstructor that was previously developed by Polites,[9] called the ideal state reconstructor. It is so called because it exactly reconstructs the state of a system during each computation cycle when the plant parameters are known exactly and the system has no process and measurement noise. In addition, the ideal state reconstructor adds no new states, eigenvalues, or dynamics to the system whose states are being estimated. In fact, it does not affect the plant equation for the system in any way; it affects the output equation only. The output of the prefilter is input into the Kalman filter each computation cycle. The first output is used to initialize the estimated state vector in the Kalman filter. This allows a fixed-gain Kalman filter to be used in place of the time-varying gain Kalman filter because the estimation errors become a minimum after the first computation cycle with either. Additional logic in the prefilter discriminates against failed or failing sensors and identifies a failed sensor in real time, much like triple-redundant sensing and voting, but with fewer sensors. Hence, this new estimation scheme has the advantages of a Kalman filter with regard to optimal estimation, but not its disadvantages.

The procedure for the presentation of this new estimation scheme and the outline for the rest of this paper is as follows. Section II reviews state estimation with Kalman filters. Section III reviews the ideal state reconstructor, a key element of the prefilter. Section IV describes the prefilter and the new estimation scheme consisting of the prefilter concatenated with a fixed-gain Kalman filter. Section V applies the new estimation scheme to an inertial plant with dual-redundant measurements of the system states. This is like a single-axis spacecraft attitude control system with dual-redundant measurements of spacecraft attitude and angular velocity. Section VI applies it to a third-order plant, also with dual-redundant measurements of the system states. This is like the single-axis spacecraft attitude control system with control actuator dynamics included, where attitude, angular velocity, and control torque are being measured redundantly. Section VII discusses strategies for applications of the scheme to plants of fourth order and higher. Section VIII offers conclusions and comments and identifies areas for future investigation.

II. Review of State Estimation with Kalman Filters

Figure 1 is a block diagram of the usual continuous-time plant driven by a zero-order hold with a sampled output. The vector $\boldsymbol{x}(t)$ is an $n \times 1$ state vector, $\boldsymbol{u}(kT)$ is an $r \times 1$ control input vector, $z(t)$ is an $m \times 1$ output vector, F is an $n \times n$ system matrix, G is an $n \times r$ control matrix, and C is an $m \times n$ output matrix. If $z(t)$ is sampled every T seconds, it is well known that this system can be modeled at the sampling instants by the discrete state equations[10]

$$\boldsymbol{x}[(k+1)T] = A\boldsymbol{x}(kT) + B\boldsymbol{x}(kT) \tag{1}$$

$$z(kT) = C\boldsymbol{x}(kT) \tag{2}$$

where

$$\phi(t) = \mathcal{L}^{-1}\lfloor (sI - F)^{-1} \rfloor \tag{3}$$

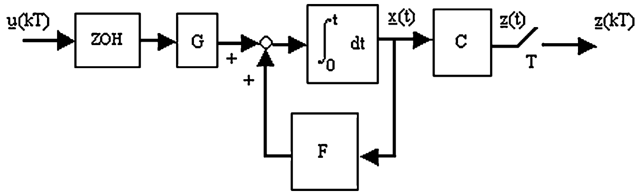

Fig. 1 Block diagram of a continuous-time plant driven by a zero-order hold with a sampled output.

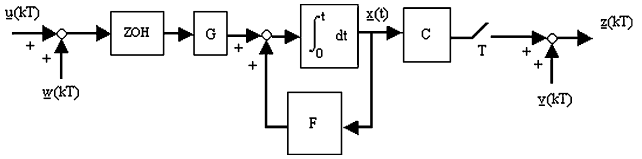

Fig. 2 Plant in Fig. 1 with process and measurement noise.

$$A = \phi(T) \tag{4}$$

$$B = \left[\int_0^T \phi(\lambda)\,\mathrm{d}\lambda\right] G \tag{5}$$

Here, $\phi(t)$ is the $n \times n$ state transition matrix and A and B are the $n \times n$ system matrix and $n \times r$ control matrix, respectively, for the discrete state Eqs. (1) and (2).

If the plant in Fig. 1 has process noise $\boldsymbol{w}(kT)$ and measurement noise $\boldsymbol{v}(kT)$ added to it, the result is the system shown in Fig. 2. The discrete state equations for modeling this system at the sampling instants are

$$\boldsymbol{x}[(k+1)T] = A\boldsymbol{x}(kT) + B\boldsymbol{u}(kT) + \boldsymbol{w}(kT) \tag{6}$$

$$z(kT) = C\boldsymbol{x}(kT) + \boldsymbol{v}(kT) \tag{7}$$

It will be assumed that the process and measurement noises are Gaussian white sequences with zero mean, are uncorrelated with each other, and have covariance matrices Q and R, respectively. These characteristics can be expressed mathematically as

$$E[\boldsymbol{w}(kT)] = E[\boldsymbol{v}(kT)] = 0 \tag{8}$$

$$E[\boldsymbol{w}(jT)\boldsymbol{w}^T(kT)] = Q\delta_{jk} \tag{9}$$

$$E[\boldsymbol{v}(jT)\boldsymbol{v}^T(kT)] = R\delta_{jk} \tag{10}$$

$$E[\boldsymbol{w}(jT)\boldsymbol{v}^T(kT)] = 0 \tag{11}$$

for all j and k.

For the system in Fig. 2 with the noise statistics as described, the Kalman filter equations for estimating the state of this system are[8,11]

$$\hat{\boldsymbol{x}}[(k+1)T/kT] = A\hat{\boldsymbol{x}}(kT/kT) + B\boldsymbol{u}(kT) \tag{12}$$

$$\hat{\boldsymbol{x}}[(k+1)T/(k+1)T] = \hat{\boldsymbol{x}}[(k+1)T/kT] + K(kT)\{z[(k+1)T] - C\hat{\boldsymbol{x}}[(k+1)T/kT]\} \tag{13}$$

In Eqs. (12) and (13), $\hat{\boldsymbol{x}}[(k+1)T/kT]$ is the predicted value of $\boldsymbol{x}$ at time $(k+1)T$ given measurements up to and including time kT, and $\hat{\boldsymbol{x}}[(k+1)T/(k+1)T]$ is the predicted value of $\boldsymbol{x}$ at time $(k+1)T$ given measurements up to and including time $(k+1)T$. The Kalman filter gain matrix, $K(kT)$, is determined by recursive solution of the equations

$$P(kT/kT) = P[kT/(k-1)T] - P[kT/(k-1)T]C^T \times \{CP[kT/(k-1)T]C^T + R\}^{-1}CP[kT/(k-1)T] \tag{14}$$

$$K(kT) = P(kT/kT)C^T R^{-1} \tag{15}$$

$$P[(k+1)T/kT] = AP(kT/kT)A^T + BQB^T \tag{16}$$

$$k = k+1 \tag{17}$$

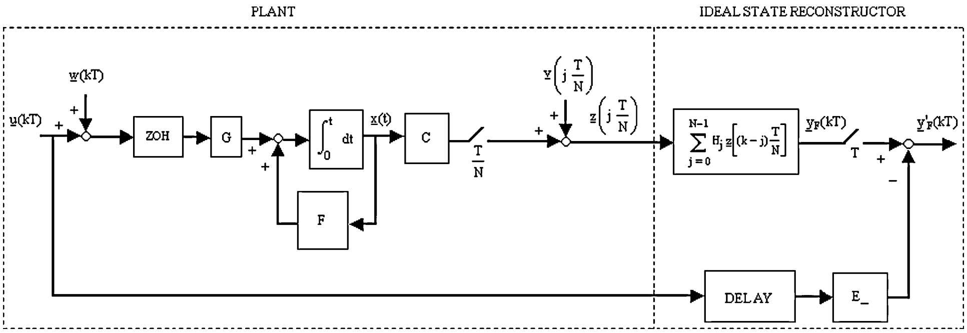
PLANT
IDEAL STATE RECONSTRUCTOR
w(kT)
u(kT)
+
+
ZOH
G
+
+
$\int_0^t dt$
x(t)
C
F
$\frac{T}{N}$
$v\left(j\frac{T}{N}\right)$
$z\left(j\frac{T}{N}\right)$
+
+
$\sum_{j=0}^{N-1} H_j z\left[(k-j)\frac{T}{N}\right]$
$y_F(kT)$
T
+
−
$y'_F(kT)$
DELAY
E_

Fig. 3 General block diagram of the plant and the ideal state reconstructor.

where, initially,

$$k = 0 \tag{18}$$

$$\hat{x}(0/0) = E[x(0)] \tag{19}$$

$$P(0/-T) = E(\{x(0) - E[x(0)]\}\{x(0) - E[x(0)]\}^T) \tag{20}$$

In Eqs. (14) and (15), $P[kT/kT]$ is the estimation error covariance matrix at time kT given measurements up to and including time kT and is given by

$$P(kT/kT) = E\lfloor\tilde{x}(kT/kT)\tilde{x}^T(kT/kT)\rfloor \tag{21}$$

where

$$\tilde{x}(kT/kT) = \hat{x}(kT/kT) - x(kT) \tag{22}$$

Equations (14–17) generate the time-varying gain matrix $K(kT)$ for the Kalman filter Eqs. (12) and (13) that minimizes $E\lfloor\tilde{x}^T(kT/kT)\tilde{x}(kT/kT)\rfloor$ for all kT. With time, $K(kT)$ converges to a steady-state matrix K_{SS} as $E\lfloor\tilde{x}^T(kT/kT)\tilde{x}(kT/kT)\rfloor$ converges to some minimum matrix. A suboptimal fixed-gain Kalman filter with gain matrix K_{SS} is sometimes used in place of the standard Kalman filter with the time-varying gains. Then, $E\lfloor\tilde{x}^T(kT/kT)\tilde{x}(kT/kT)\rfloor$ still converges to the same minimum matrix than it does with the optimal time-varying gain matrix, but it takes longer.

III. Review of the Ideal State Reconstructor

Figure 3 is a general block diagram of the ideal state reconstructor concatenated with the plant of Fig. 2, but with $z(t)$ sampled every T/N seconds instead of every T seconds. These discrete measurements are input into a multi-input/multi-output moving average (MA) filter[12] with coefficient matrices H_j, $j = 0, 1, \ldots, N-1$, and the output is sampled every T seconds to generate $y_F(kT)$. Recognize that this is equivalent to taking N discrete measurements in a T-second interval, multiplying each by an appropriate weighting matrix, and then summing them recursively as they are generated to produce $y_F(kT)$. Then $y_F(kT)$ has subtracted from it $E_{-}u[(k-1)T]$, where E_{-} is a constant matrix, to produce the output vector $y'_F(kT)$. Let H_j, $j = 0, 1, \ldots, N-1$, each be an $n \times m$ matrix and E_{-} be an $n \times r$ matrix. Then $y_F(kT)$ and $y'_F(kT)$ are both $n \times 1$ vectors. Polites[9] showed that if N; H_j, $j = 0, 1, \ldots, N-1$; and E_{-} are appropriately chosen, the entire system in Fig. 3 can be modeled at the sampling instants by the discrete state equations

$$x[(k+1)T] = Ax(kT) + Bu(kT) \tag{23}$$

$$y'_F(kT) = x(kT) \tag{24}$$

This means that $y'_F(kT)$ exactly reconstructs the state vector $x(kT)$, and the reconstruction process has not affected the plant equation for the system in any way.

The procedure for choosing N; H_j, $j = 0, 1, \ldots, N-1$; and E_{-} to achieve these results is as follows. First, let

$$N \geq n/m \tag{25}$$

Now let

$$\alpha = \begin{bmatrix} C\phi(0) \\ C\phi\left(-\frac{T}{N}\right) \\ \cdots \\ \cdots \\ \cdots \\ C\phi\left(-(N-1)\frac{T}{N}\right) \end{bmatrix} \tag{26}$$

$$H = (\alpha^T\alpha)^{-1}\alpha^T \tag{27}$$

Note that α and H are $(Nm) \times n$ and $n \times (Nm)$ matrices, respectively. If $N \geq n/m$ and $\mathrm{rank}(\alpha) = n$, then $(\alpha^T\alpha)$ is positive definite and, therefore, nonsingular, and the pseudoinverse of α, given by Eq. (27), exists.[13] Under the assumption that this is the case, to determine H_j, $j = 0, 1, \ldots, N-1$, partition H into

$$H = [H_0 : H_1 : \ldots : H_{N-1}] \tag{28}$$

Recognize that each H_j is an $n \times m$ matrix. To determine E_{-}, let

$$\beta = \begin{bmatrix} C\left[\int_0^0 \phi(\lambda)\,\mathrm{d}\lambda\right]G \\ C\left[\int_0^{-(T/N)} \phi(\lambda)\,\mathrm{d}\lambda\right]G \\ \cdots \\ \cdots \\ \cdots \\ C\left[\int_0^{-(N-1)(T/N)} \phi(\lambda)\,\mathrm{d}\lambda\right]G \end{bmatrix} \tag{29}$$

where β is an $(Nm) \times r$ matrix, and

$$E_{-} = H\beta \tag{30}$$

This completely defines the parameters in the ideal state reconstructor and assures that the entire system in Fig. 3 can be modeled at the sampling instants by the discrete state equations (23) and (24).

IV. Description of the Prefilter and the New Estimation Scheme Consisting of the Prefilter Concatenated with a Fixed-Gain Kalman Filter

The prefilter utilizes the ideal state reconstructor to generate preliminary estimates of the system state vector during each computation cycle, to discriminate against failed and failing sensors, and to identify a failed sensor in just one computation cycle. How the prefilter accomplishes this is best described with an example, which will

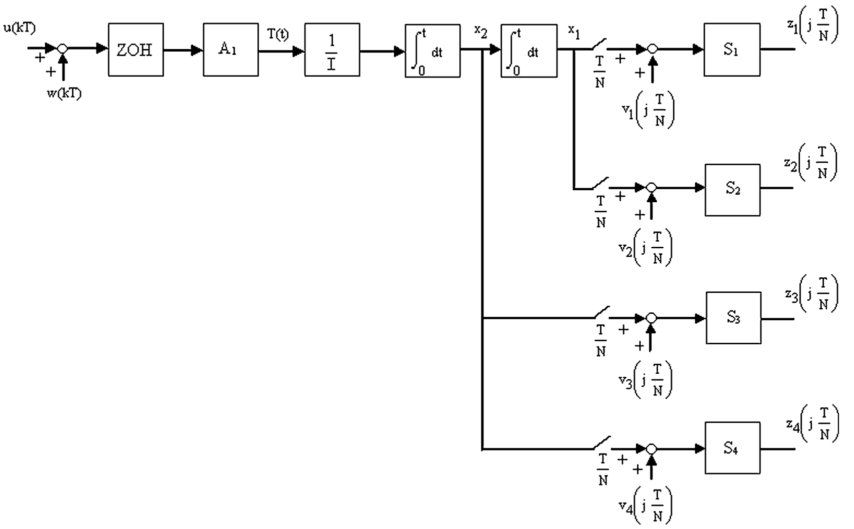

Fig. 4 Inertial plant with dual-redundant measurements of the system states.

be an inertial plant with dual-redundant measurements of the system states (Fig. 4). This is like a single-axis spacecraft attitude control system, in which the spacecraft moment of inertia is I. Discrete torque commands, $u(kT)$, are issued from an onboard computer every T seconds. By virtue of the zero-order hold and the control actuator, designated A_1, torque is applied to the spacecraft continuously. Normally, $A_1 = 1$. If the actuator fails, then $A_1 = 0$. The attitude sensors are designated by S_1 and S_2, whereas the angular velocity sensors are designated by S_3 and S_4. Normally, $S_1 = S_2 = S_3 = S_4 = 1$; but if a sensor fails, the corresponding S_i is set equal to 0. Measurements from the sensors are generated every T/N seconds and are corrupted by noise as shown.

For the system in Fig. 4, the prefilter will be required to generate an estimate of the system state vector every T seconds, in spite of a failure of any one of the four sensors. It is improbable that any two sensors will fail at the same time. If a sensor does fail, the prefilter must still generate a correct estimate of the system state vector without isolating the failed sensor or reconfiguring the estimation algorithms, and the failed sensor should be identified immediately. These are the requirements on the prefilter.

To meet these requirements, the prefilter utilizes the ideal state reconstructor. Recognize that the ideal state reconstructor can estimate the system state vector every T seconds in a number of different ways using various sensor combinations. For example, it can estimate the system state vector using only one position sensor, $S1$ or $S2$; it can estimate the system state vector using one position sensor, $S1$ or $S2$, and one rate sensor, $S3$ or $S4$; and so on. In fact, it can estimate the system state vector using any combination of sensors that yields an observable system. However, the prefilter will be configured to estimate the system state vector using the ideal state reconstructor and the following sensor combinations: $S1$; $S2$; $S1$–$S3$; $S1$–$S4$; $S2$–$S3$; $S2$–$S4$. This is the set of all combinations of one position sensor or one position sensor and one rate sensor. In a sense, this is like measuring the system state vector with six unique virtual state vector sensors.

Now, if the plant parameter I is precisely known, the system has no process noise, and the sensors have no measurement noise, then the estimated state vectors generated every T seconds using the ideal state reconstructor and these sensor combinations will exactly equal the true system state vector. On the other hand, if the plant parameter I is approximately known, the system has some process noise, and the sensors have some measurement noise, then all of the estimated state vectors generated every T seconds with the ideal state reconstructor will be approximations to the true system state vector. That is, they will be close to the true system state vector and, equally important, close to each other. Thus, the estimated state vectors will tend to cluster at every T-second interval.

Now, if a sensor fails at some point in time, the estimated state vectors generated with good sensors will still cluster, but those generated with the failed sensor will deviate from this cluster. For example, suppose S_1 fails, then the estimated state vector using sensors S_2, S_2–S_3, and S_2–S_4 will cluster and those using sensors S_1, S_1–S_3, and S_1–S_4 will deviate from the cluster. If S_3 fails, then the estimated state vectors using sensors S_1, S_2, S_1–S_4, and S_2–S_4 will cluster, and those using sensors S_1–S_3 and S_2–S_3 will deviate. Thus, if a sensor fails, at least three estimated state vectors are assured to cluster. Identification of the three that are the most tightly clustered and determination of the centroid of these provides a good estimate of the system state vector despite a sensor failure. This is the process of generation of a preliminary estimate of the system state vector and discrimination against a failed sensor. Determination of which, if any, estimated state vectors deviate significantly from this centroid and the relation of these to the sensors used to generate those estimated state vectors provides sufficient information to identify the failed sensor immediately. This is how the prefilter identifies a failed sensor in real time. Conceptually, this is similar in some ways to state estimation with triple-redundant sensing and sensor voting, but with fewer sensors.

To determine the three estimated state vectors that are the most tightly clustered, the elements of each are first normalized with respect to some expected maximum to give each element equal weight. That is, if the estimated state vectors generated with the ideal state reconstructor and sensors $S2$, $S2$–$S3$, and $S2$–$S4$, are, respectively,

$$y'_{F2}(kT) = \begin{bmatrix} \hat{x}_{1/2}(kT) \\ \hat{x}_{2/2}(kT) \end{bmatrix}, \qquad y'_{F23}(kT) = \begin{bmatrix} \hat{x}_{1/23}(kT) \\ \hat{x}_{2/23}(kT) \end{bmatrix}$$

$$y'_{F24}(kT) = \begin{bmatrix} \hat{x}_{1/24}(kT) \\ \hat{x}_{2/24}(kT) \end{bmatrix} \tag{31}$$

then the corresponding normalized estimated state vectors are

$$y'_{F2/N}(kT) = \begin{bmatrix} \dfrac{\hat{x}_{1/2}(kT)}{x_{1\max}} \\[2ex] \dfrac{\hat{x}_{2/2}(kT)}{x_{2\max}} \end{bmatrix}, \qquad y'_{F23/N}(kT) = \begin{bmatrix} \dfrac{\hat{x}_{1/23}(kT)}{x_{1\max}} \\[2ex] \dfrac{\hat{x}_{2/23}(kT)}{x_{2\max}} \end{bmatrix}$$

$$y'_{F24/N}(kT) = \begin{bmatrix} \dfrac{\hat{x}_{1/24N}(kT)}{x_{1\max}} \\[2ex] \dfrac{\hat{x}_{2/24N}(kT)}{x_{2\max}} \end{bmatrix} \tag{32}$$

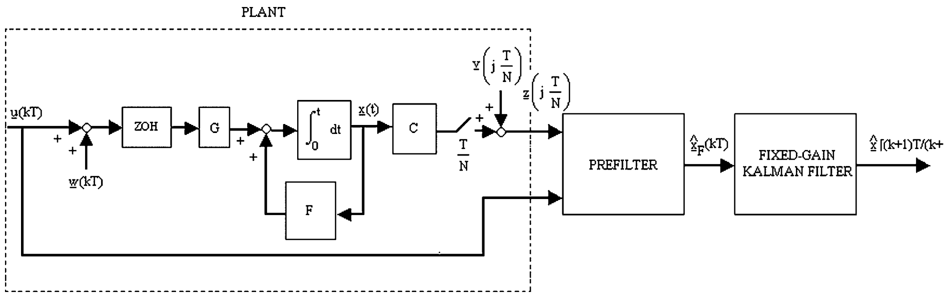

Fig. 5 New estimation scheme consisting of the prefilter concatenated with a fixed-gain Kalman filter.

Next, the relative differences in these normalized estimated state vectors are computed. Then, the norms of these differences are determined and summed for this triad. The process is repeated for the other triads, and the minimum sum gives the triad that is the most tightly clustered. The centroid of this triad is the prefilter's normalized estimate for the state of the system at that point in time. It will be denoted by $\hat{\boldsymbol{x}}_N(kT)$. Multiplication of each component of this vector by the appropriate normalization factor yields the estimated state vector $\hat{\boldsymbol{x}}(kT)$ output from the prefilter and input into the Kalman filter at each computation cycle.

To determine if a sensor has failed, the six normalized estimated state vectors generated inside the prefilter with the ideal state reconstructor, $\boldsymbol{y}'_{F1/N}(kT), \boldsymbol{y}'_{F2/N}(kT), \boldsymbol{y}'_{F13/N}(kT), \boldsymbol{y}'_{F14/N}(kT)$, $\boldsymbol{y}'_{F23/N}(kT)$, and $\boldsymbol{y}'_{F24/N}(kT)$, are differenced with $\hat{x}_N(kT)$. The norms of the differences are then compared with some specified threshold ε.

Then, if

$$\|\boldsymbol{y}'_{F2/N}(kT) - \hat{\boldsymbol{x}}_N(kT)\| \leq \varepsilon, \qquad \|\boldsymbol{y}'_{F23/N}(kT) - \hat{\boldsymbol{x}}_N(kT)\| \leq \varepsilon$$

$$\|\boldsymbol{y}'_{F24/N}(kT) - \hat{\boldsymbol{x}}_N(kT)\| \leq \varepsilon \tag{33}$$

$$\|\boldsymbol{y}'_{F1/N}(kT) - \hat{\boldsymbol{x}}_N(kT)\| \geq \varepsilon, \qquad \|\boldsymbol{y}'_{F13/N}(kT) - \hat{\boldsymbol{x}}_N(kT)\| \geq \varepsilon$$

$$\|\boldsymbol{y}'_{F14/N}(kT) - \hat{\boldsymbol{x}}_N(kT)\| \geq \varepsilon \tag{34}$$

it is concluded that sensor S_1 has failed. If

$$\|\boldsymbol{y}'_{F1/N}(kT) - \hat{\boldsymbol{x}}_N(kT)\| \leq \varepsilon, \qquad \|\boldsymbol{y}'_{F13/N}(kT) - \hat{\boldsymbol{x}}_N(kT)\| \leq \varepsilon$$

$$\|\boldsymbol{y}'_{F14/N}(kT) - \hat{\boldsymbol{x}}_N(kT)\| \leq \varepsilon \tag{35}$$

$$\|\boldsymbol{y}'_{F2/N}(kT) - \hat{\boldsymbol{x}}_N(kT)\| \geq \varepsilon, \qquad \|\boldsymbol{y}'_{F23/N}(kT) - \hat{\boldsymbol{x}}_N(kT)\| \geq \varepsilon$$

$$\|\boldsymbol{y}'_{F24/N}(kT) - \hat{\boldsymbol{x}}_N(kT)\| \geq \varepsilon \tag{36}$$

it is concluded that sensor S_2 has failed. If

$$\|\boldsymbol{y}'_{F1/N}(kT) - \hat{\boldsymbol{x}}_N(kT)\| \leq \varepsilon, \qquad \|\boldsymbol{y}'_{F2/N}(kT) - \hat{\boldsymbol{x}}_N(kT)\| \leq \varepsilon$$

$$\|\boldsymbol{y}'_{F14/N}(kT) - \hat{\boldsymbol{x}}_N(kT)\| \leq \varepsilon, \qquad \|\boldsymbol{y}'_{F24/N}(kT) - \hat{\boldsymbol{x}}_N(kT)\| \leq \varepsilon \tag{37}$$

$$\|\boldsymbol{y}'_{F13/N}(kT) - \hat{\boldsymbol{x}}_N(kT)\| \geq \varepsilon, \qquad \|\boldsymbol{y}'_{F23/N}(kT) - \hat{\boldsymbol{x}}_N(kT)\| \geq \varepsilon \tag{38}$$

it is concluded that sensor S_3 has failed. If

$$\|\boldsymbol{y}'_{F1/N}(kT) - \hat{\boldsymbol{x}}_N(kT)\| \leq \varepsilon, \qquad \|\boldsymbol{y}'_{F2/N}(kT) - \hat{\boldsymbol{x}}_N(kT)\| \leq \varepsilon$$

$$\|\boldsymbol{y}'_{F13/N}(kT) - \hat{\boldsymbol{x}}_N(kT)\| \leq \varepsilon, \qquad \|\boldsymbol{y}'_{F23/N}(kT) - \hat{\boldsymbol{x}}_N(kT)\| \leq \varepsilon \tag{39}$$

$$\|\boldsymbol{y}'_{F14/N}(kT) - \hat{\boldsymbol{x}}_N(kT)\| \geq \varepsilon, \qquad \|\boldsymbol{y}'_{F24/N}(kT) - \hat{\boldsymbol{x}}_N(kT)\| \geq \varepsilon \tag{40}$$

it is concluded that sensor S_4 has failed.

The new estimation scheme consists of the prefilter, just described, concatenated with a fixed-gain Kalman filter, as shown in Fig. 5. The measurements from the plant are input into the prefilter every T/N seconds. The output of the prefilter is a preliminary estimate of the system state every T seconds. This is input into a fixed-gain Kalman filter. The fixed-gain Kalman filter can be used in place of the standard time-varying gain Kalman filter because the prefilter generates a good estimate of the system state vector after just one computation cycle. This can be used to initialize the estimated state vector in the Kalman filter. Consequently, the estimation errors in the Kalman filter are at a minimum after the first computation cycle, allowing the fixed-gain Kalman filter to be used in place of the time-varying gain Kalman filter. By virtue of the plant concatenated with the prefilter, Eqs. (23) and (24) become the plant equations for designing the Kalman filter.

V. New Estimation Scheme Applied to an Inertial Plant with Dual Redundant Measurements of the System States

The new estimation scheme described in Sec. IV was tested on the plant in Fig. 4, which is like a single-axis spacecraft attitude control system. A computer simulation of the entire system was developed. The plant parameters were chosen to be

$$I = 10^4\ \mathrm{kg \cdot m^2}, \qquad T = 0.1\ \mathrm{s}, \qquad N = 10 \tag{41}$$

The sensor and process noises were assumed to be Gaussian white sequences with zero mean and

$$v_1(jT/N)|_{1\sigma} = v_2(jT/N)|_{1\sigma} = \pi \times 10^{-4}\ \mathrm{rad} \tag{42}$$

$$v_3(jT/N)|_{1\sigma} = v_4(jT/N)|_{1\sigma} = 10^{-4}\ \mathrm{rad/s} \tag{43}$$

$$\boldsymbol{w}(kT)|_{1\sigma} = 0.1\ \mathrm{N \cdot m} \tag{44}$$

These are values the new scheme can tolerate, plus they are readily achievable with spacecraft of today.

In the prefilter,

$$x_{1\,\mathrm{max}} = \pi\ \mathrm{rad}, \qquad x_{2\,\mathrm{max}} = 0.1\ \mathrm{rad/s} \tag{45}$$

A good value for the threshold to identify a failed sensor was found by simulation to be

$$\varepsilon = 1.75 \tag{46}$$

To determine the steady-state Kalman filter gain matrix K_{SS}, the rms errors in the estimated states out of the prefilter were computed over a period of time in the simulation, and these were used in the R matrix of the Kalman filter equations. It turns out that R was set equal to

$$R = \begin{bmatrix} (8.87 \times 10^{-5}\ \mathrm{rad})^2 & 0 \\ 0 & (113.0 \times 10^{-5}\ \mathrm{rad/s})^2 \end{bmatrix} \tag{47}$$

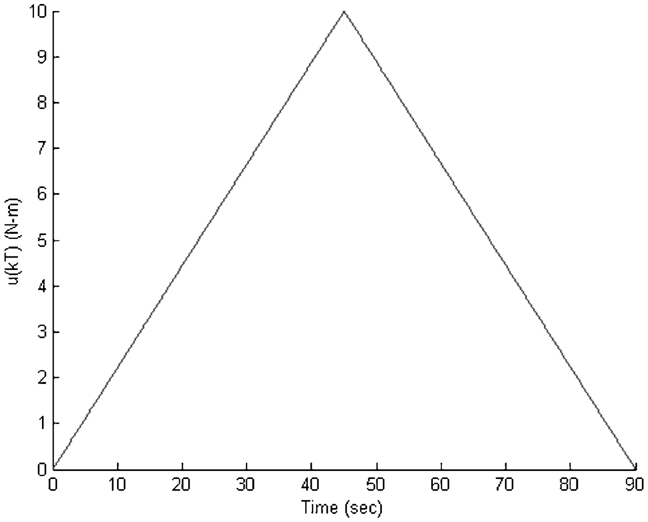

Fig. 6 Discrete torque command profile $u(kT)$ vs time for the simulation test cases.

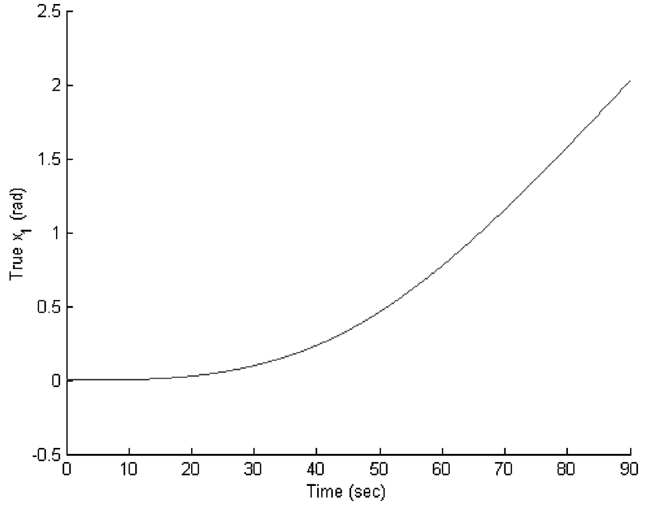

Fig. 7 True state $x_1(kT)$ vs time for the simulation test cases.

Q and $P(0/-T)$ were chosen to be

$$Q = (0.1\ \mathrm{N\cdot m})^2 \tag{48}$$

$$P(0/-T) = \begin{bmatrix} (\pi\ \mathrm{rad})^2 & 0 \\ 0 & (0.1\ \mathrm{rad/s})^2 \end{bmatrix} \tag{49}$$

respectively. This gave

$$K_{ss} = \begin{bmatrix} 4.64\times10^{-2} & 6.78\times10^{-5} \\ 1.10\times10^{-2} & 3.26\times10^{-5} \end{bmatrix} \tag{50}$$

The entire system was simulated with the initial conditions $x_1(0) = x_2(0) = 0$ and the discrete torque command profile shown in Fig. 6. For this command profile, the true states of the system vs time are shown in Figs. 7 and 8. The initial estimated states for the Kalman filter were the estimated states from the prefilter after the first computation cycle. Two cases were simulated.

The first case was with S_1 failed at $t = 45$ s. The prefilter discriminated against the failed sensor and successfully identified that S_1 had failed at $t = 45$ s. The estimation errors vs time from the prefilter and the Kalman filter are shown in Figs. 9 and 10. The rms of the estimation errors from the prefilter and the Kalman filter computed over the entire 90-s simulation run are shown in Table 1.

Note that the Kalman filter reduced the estimation errors from the prefilter by about a factor of 5 for $\tilde{x}_1(kT)$ and a factor of 200 for $\tilde{x}_2(kT)$. When the noise in the measurements is compared with the outputs of the Kalman filter, the entire estimation scheme reduced the estimation errors about a factor of 10 for $\tilde{x}_1(kT)$ and about a factor of 20 for $\tilde{x}_2(kT)$.

Table 1 Estimation errors for new estimation scheme with S_1 failed at t = 45 s

Simulation results	rms of $\tilde{x}_1(kT)$, rad	rms of $\tilde{x}_2(kT)$, rad/s
Prefilter output	10.8×10^{-5}	113.0×10^{-5}
Kalman filter output	2.16×10^{-5}	0.48×10^{-5}

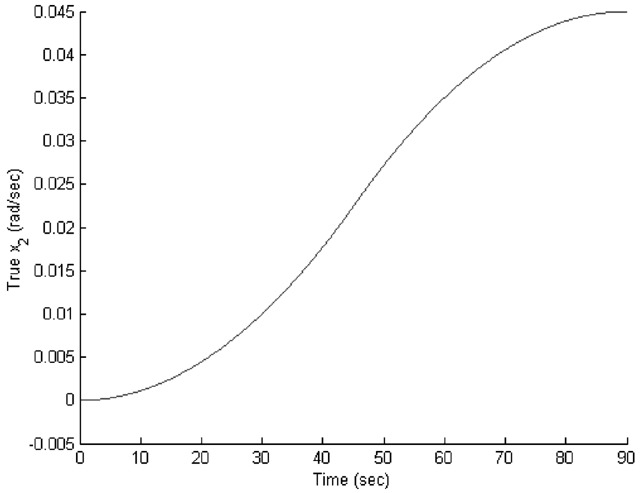

Fig. 8 True state $x_2(kT)$ vs time for the simulation test cases.

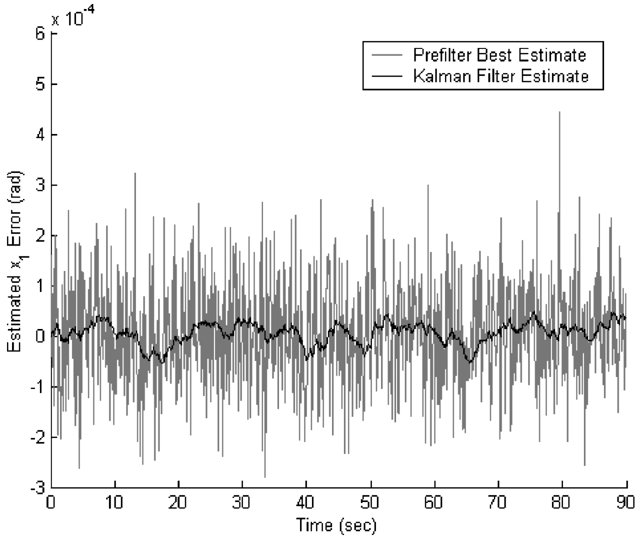

Fig. 9 Estimation error $\tilde{x}_1(kT)$ vs time with the inertial plant, the new estimation scheme, and S_1 failed at $t = 45$ s.

The second case was with S_3 failed at $t = 45$ s. The prefilter discriminated against the failed sensor and successfully identified that S_3 had failed at $t = 45$ s. The estimation errors vs time were similar to those in Figs. 9 and 10. The rms of the estimation errors from the prefilter and the Kalman filter computed over the entire 90-s simulation run are shown in Table 2.

As before, the Kalman filter reduced the estimation errors from the prefilter by about a factor of 5 for $\tilde{x}_1(kT)$ and a factor of 200 for $\tilde{x}_2(kT)$. When the noise in the measurements is compared with the outputs of the Kalman filter, the entire estimation scheme again reduced the estimation errors about a factor of 10 for $\tilde{x}_1(kT)$ and about a factor of 20 for $\tilde{x}_2(kT)$.

Table 2 Estimation errors for new estimation scheme with S_3 failed at t = 45 s

Simulation results	rms of $\tilde{x}_1(kT)$, rad	rms of $\tilde{x}_2(kT)$, rad/s
Prefilter output	8.9×10^{-5}	113.0×10^{-5}
Kalman filter output	1.87×10^{-5}	0.41×10^{-5}

Table 3 Steady-state estimation errors with just the time-varying gain Kalman filter

Simulation results	Kalman filter output
rms of $\tilde{x}_1(kT)$, rad	4.18×10^{-5}
rms of $\tilde{x}_2(kT)$, rad/s	1.14×10^{-5}

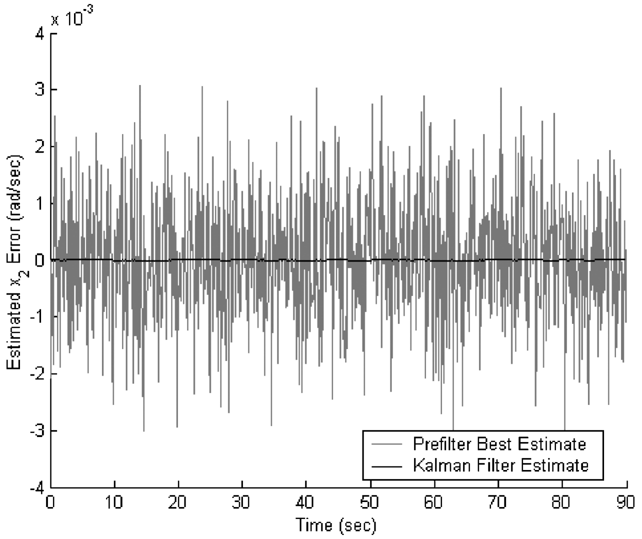

Fig. 10 Estimation error $\tilde{x}_2(kT)$ vs time with the inertial plant, the new estimation scheme, and S_1 failed at $t = 45$ s.

To compare the new estimation scheme with the Kalman filter by itself, the states for the plant in Fig. 4 were estimated with just the standard time-varying gain Kalman filter. It was assumed that the fastest that the Kalman filter equations could be solved was once every 0.1 s. Thus, measurements every 0.1 s were input into the Kalman filter and the other measurements were discarded. One might argue that today's computers could solve the Kalman filter equations more frequently than this. True, but whatever the limiting frequency, it is also likely that today's sensors could generate measurements at a faster rate than this. Hence, this should be a fair test of the performance of the new estimation scheme compared to the Kalman filter by itself. With this scenario and no sensor failures over the entire 90-s simulation, the steady-state estimation errors with just the time-varying gain Kalman filter are those in Table 3.

This shows that the new estimation scheme out performed the standard time-varying gain Kalman filter by about a factor of two.

Next, the same case was run, but with sensor S_1 failed at $t = 45$ s. This case was then repeated with S_3 failed at $t = 45$ s instead. In both cases, the Kalman filter could not generate good estimates for the system states after the sensor failure, as expected.

Then, the same set of simulation runs were made with just the fixed-gain Kalman filter. With no sensor failures, the steady-state estimation errors are shown in Table 4.

Again, the new estimation scheme out performed the fixed-gain Kalman filter by about a factor of two, as expected. With sensor failures, the fixed-gain Kalman filter could not generate good estimates for the system states, also as expected.

Table 4 Steady-state estimation errors with just the fixed-gain Kalman filter

Simulation results	Kalman filter output
rms of $\tilde{x}_1(kT)$, rad	4.45×10^{-5}
rms of $\tilde{x}_2(kT)$, rad/s	1.21×10^{-5}

VI. New Estimation Scheme Applied to a Third-Order Plant with Dual-Redundant Measurements of the System States

Next, the new estimation scheme was applied to the plant in Fig. 11, which is a third-order system with dual-redundant measurements of the system states. This is like the plant in Fig. 4 with control actuator dynamics included and modeled by a first-order lag.

For this system, the prefilter is configured as before, except that now (S_1–S_3) and (S_2–S_4) are treated as units, and the prefilter estimates the system state vectors every T seconds using sensor combinations: (S_1–S_3), (S_2–S_4), (S_1–S_3)–S_5, (S_1–S_3)–S_6, (S_2–S_4)–S_5, (S_2–S_4)–S_6. The three estimated state vectors that are the most tightly clustered are found as before, and the centroid of this triad is the output of the prefilter for that computation cycle. Deviations of the estimated state vectors from the centroid allow identification of a failure in either (S_1–S_3), (S_2–S_4), S_5, or S_6. If a failure is identified with (S_1–S_3) or (S_2–S_4), the prefilter determines the specific sensor or sensors that failed by comparing the average outputs from S_1 and S_2 over the last T and the average outputs from S_3 and S_4 over the last T seconds. If the magnitude of the difference in the average outputs from S_1 and S_2 over the last T seconds exceeds some predetermined threshold ε_{1-2}, and the prefilter identifies a failure in S_1–S_3, then it concludes that S_1 has failed; if it identifies a failure in S_2–S_4, then it concludes that S_2 has failed. In like manner, if the magnitude of the difference in the average outputs from S_3 and S_4 over the last T seconds exceeds some predetermined threshold ε_{3-4}, and the prefilter identifies a failure in S_1–S_3, then the prefilter concludes that S_3 has failed; if the prefilter identifies a failure in S_2–S_4, then it concludes that S_4 has failed. The rest of the estimation scheme is the same as before. That is, the output of the prefilter is input into the fixed-gain Kalman filter and the fixed-gain Kalman filter assumes the plant model described by Eqs. (23) and (24).

A simulation of the entire system was developed as before, and the same plant parameters were used. The parameter associated with the actuator dynamics was chosen to be

$$\omega_a = 2\pi \text{ rad/s} \tag{51}$$

The sensor and process noises were again assumed to be Gaussian white sequences with zero mean and

$$v_1(jT/N)|_{1\sigma} = v_2(jT/N)|_{1\sigma} = \pi \times 10^{-4} \text{ rad} \tag{52}$$

$$v_3(jT/N)|_{1\sigma} = v_4(jT/N)|_{1\sigma} = 10^{-5} \text{ rad/s} \tag{53}$$

$$v_5(jT/N)|_{1\sigma} = v_6(jT/N)|_{1\sigma} = 0.01 \text{ N}\cdot\text{m} \tag{54}$$

$$w(kT)|_{1\sigma} = 0.01 \text{ N}\cdot\text{m} \tag{55}$$

These are values the new scheme can tolerate, plus they are still achievable with spacecraft of today.

In the prefilter,

$$x_{1\,\max} = \pi \text{ rad}, \qquad x_{2\,\max} = 0.1 \text{ rad/s}, \qquad x_{3\,\max} = 20 \text{ N}\cdot\text{m} \tag{56}$$

$$\varepsilon = 2.5, \qquad \varepsilon_{1-2} = 0.03 \text{ rad}, \qquad \varepsilon_{3-4} = 0.001 \text{ rad/s} \tag{57}$$

To determine the steady-state Kalman filter gain matrix K_{SS}, the R matrix was determined as it was before,

$$Q = (0.01 \text{ Nm})^2 \tag{58}$$

$$P(0/-T) = \begin{bmatrix} (\pi \text{ rad})^2 & 0 & 0 \\ 0 & (0.1 \text{ rad/s})^2 & 0 \\ 0 & 0 & (20 \text{ Nm})^2 \end{bmatrix} \tag{59}$$

This gave

$$K_{ss} = \begin{bmatrix} 4.91\times10^{-3} & 8.21\times10^{-2} & -1.04\times10^{-10} \\ 2.05\times10^{-4} & 2.39\times10^{-2} & 5.44\times10^{-9} \\ -1.47\times10^{-3} & 3.06\times10^{1} & 3.38\times10^{-4} \end{bmatrix} \tag{60}$$

The simulation was run for $T_{\max} = 90$ s with the initial conditions $x_1(0) = x_2(0) = x_3(0) = 0$ and the discrete torque command profile $u(kT)$ shown in Fig. 6. The initial estimated states for the Kalman filter were the estimated states from the prefilter after the first computation cycle. Three cases were simulated.

The first case was with S_1 failed at $t = 45$ s. The prefilter discriminated against the failed sensor and successfully identified that S_1 failed at $t = 45$ s. The estimation errors vs time from the prefilter and the Kalman filter are shown in Figs. 12–14. The rms of the estimation errors from the prefilter and the Kalman filter computed over the entire 90-s simulation run are shown in Table 5.

Note that the Kalman filter reduced the estimation errors from the prefilter by about a factor of 15 for $\tilde{x}_1(kT)$, a factor of 10 for $\tilde{x}_2(kT)$, and a factor of 160 for $\tilde{x}_3(kT)$.

The second case was with S_3 failed at $t = 45$ s. The prefilter discriminated against the failed sensor and successfully identified that S_3 had failed at $t = 45$ s. The estimation errors vs time from the prefilter and the Kalman filter are similar to those shown in Figs. 12–14. The rms of the estimation errors from the prefilter and the Kalman filter computed over the entire 90-s simulation run are shown in Table 6.

Table 5 Estimation errors for new estimation scheme with S_1 failed at $t = 45$ s

Simulation results	rms of $\tilde{x}_1(kT)$, rad	rms of $\tilde{x}_2(kT)$, rad/s	rms of $\tilde{x}_3(kT)$, N · m
Prefilter output	96.2×10^{-6}	45.5×10^{-7}	286.0×10^{-3}
Kalman filter output	6.43×10^{-6}	4.66×10^{-7}	1.79×10^{-3}

Again, the Kalman filter reduced the estimation errors from the prefilter by about a factor of 15 for $\tilde{x}_1(kT)$, a factor of 10 for $\tilde{x}_2(kT)$, and a factor of 160 for $\tilde{x}_3(kT)$.

The third case was with S_5 failed at $t = 45$ s. The prefilter discriminated against the failed sensor and successfully identified that S_5 had failed at $t = 45$ s. The estimation errors vs time from the

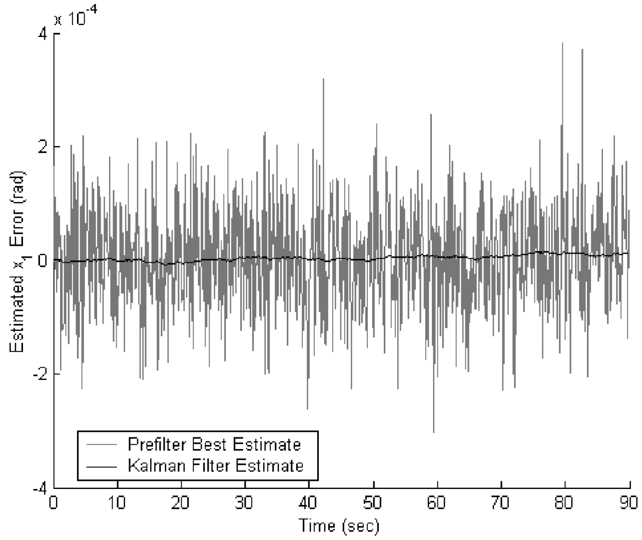

Fig. 12 Estimation error $\tilde{x}_1(kT)$ vs time with the third-order plant, the new estimation scheme, and S_1 failed at $t = 45$ s.

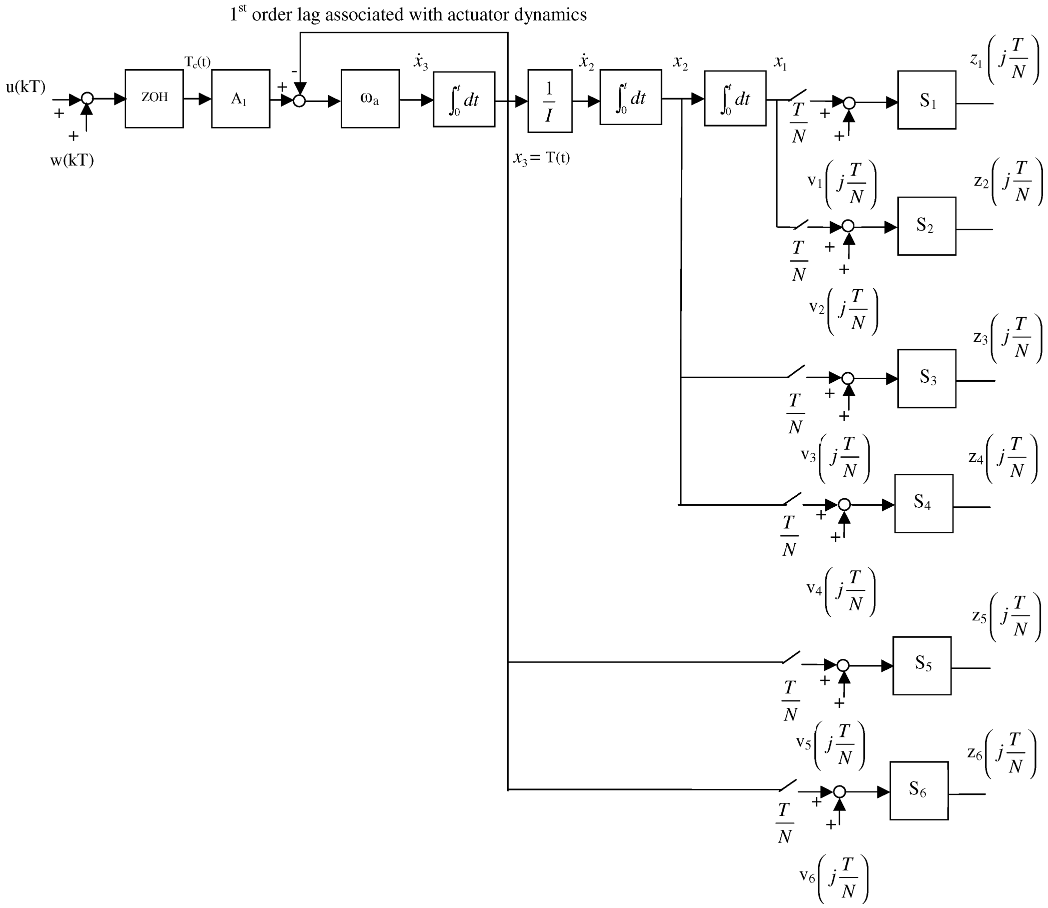

Fig. 11 Third-order plant with dual-redundant measurements of the system states.

Table 6 Estimation errors for new estimation scheme with S_3 failed at $t = 45$ s

Simulation results	rms of $\tilde{x}_1(kT)$, rad	rms of $\tilde{x}_2(kT)$, rad/s	rms of $\tilde{x}_3(kT)$, N · m
Prefilter output	96.2×10^{-6}	45.5×10^{-7}	286.0×10^{-3}
Kalman filter output	6.43×10^{-6}	4.66×10^{-7}	1.79×10^{-3}

Table 7 Estimation errors for new estimation scheme with S_5 failed at $t = 45$ s

Simulation results	rms of $\tilde{x}_1(kT)$, rad	rms of $\tilde{x}_2(kT)$, rad/s	rms of $\tilde{x}_3(kT)$, N · m
Prefilter output	74.8×10^{-6}	35.0×10^{-7}	286.0×10^{-3}
Kalman filter output	8.75×10^{-6}	4.09×10^{-7}	1.80×10^{-3}

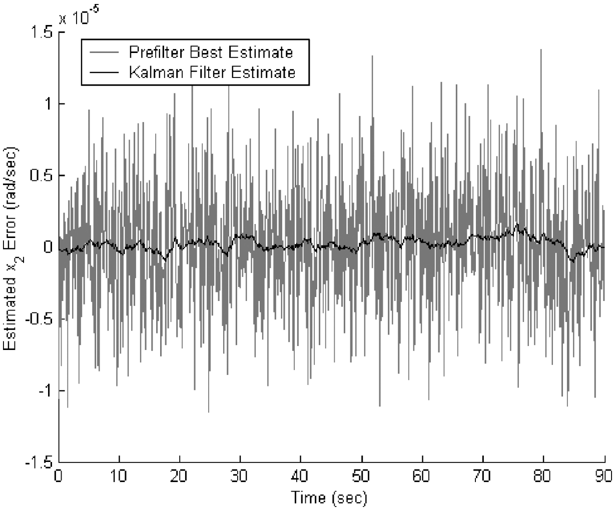

Fig. 13 Estimation error $\tilde{x}_2(kT)$ vs time with the third-order plant, the new estimation scheme, and S_1 failed at $t = 45$ s.

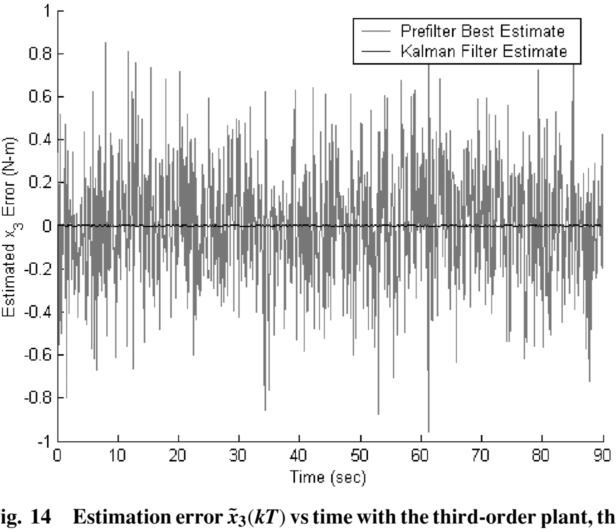

Fig. 14 Estimation error $\tilde{x}_3(kT)$ vs time with the third-order plant, the new estimation scheme, and S_1 failed at $t = 45$ s.

prefilter and the Kalman filter are again similar to those shown in Figs. 12–14. The rms of the estimation errors from the prefilter and the Kalman filter computed over the entire 90-s simulation run are shown in Table 7.

Now, the Kalman filter reduced the estimation errors from the prefilter by about a factor of 9 for $\tilde{x}_1(kT)$, a factor of 9 for $\tilde{x}_2(kT)$, and a factor of 159 for $\tilde{x}_3(kT)$.

As before, simulation runs were made with just the time-varying gain Kalman filter and then just the fixed-gain Kalman filter for comparisons. The results are shown in Tables 8 and 9, respectively.

Table 8 Steady-state estimation errors with just the time-varying gain Kalman filter

Simulation results	Kalman filter output
rms of $\tilde{x}_1(kT)$, rad	36.5×10^{-6}
rms of $\tilde{x}_2(kT)$, rad/s	40.3×10^{-7}
rms of $\tilde{x}_3(kT)$, N · m	4.3×10^{-3}

Table 9 Steady-state estimation errors with just the fixed-gain Kalman filter

Simulation results	Kalman filter output
rms of $\tilde{x}_1(kT)$, rad	36.6×10^{-6}
rms of $\tilde{x}_2(kT)$, rad/s	40.4×10^{-7}
rms of $\tilde{x}_3(kT)$, N · m	4.31×10^{-3}

Tables 8 and 9 show that the new estimation scheme outperformed by about a factor of 4 with regard to $\tilde{x}_1(kT)$, a factor of 10 with regard to $\tilde{x}_2(kT)$, and a factor of 2 with regard to $\tilde{x}_3(kT)$. When sensors failed, the Kalman filters alone could not generate good estimates for the system states, as expected.

A variation of the plant in Fig. 11 was also investigated. Here, state x_3 was not measured, but was estimated in the prefilter and the fixed-gain Kalman filter. In this case, the approach is just like that in Sec. V, only now the estimated state vectors in the prefilter and the fixed-gain Kalman filter have three components instead of two. It was found that the new estimation scheme performed well, but less sensor noise could be tolerated. This is not surprising because effectively two differentiations are required to estimate x_3 by the use of the ideal state reconstructor and measurements from sensors S_1 or S_2. One differentiation is required when sensors S_1–S_3, S_1–S_4, S_2–S_3, or S_2–S_4. are used. Differentiating noise greatly amplifies it; hence, less can be tolerated. From this experience, it appears best to estimate the same number of states that are being measured. Certainly, this is an area for further study.

VII. New Estimation Scheme Applied to Plants of Fourth Order and Higher

Suppose the new estimation scheme is to be applied to a fourth-order plant with states x_1, x_2, x_3, and x_4. With dual-redundant measurements of the states x_1 and x_2, the scheme described in Sec. V could be used, only now there are four states to be estimated instead of two. Anticipate that less sensor noise can be tolerated. With dual-redundant measurements of x_1, x_2, and x_3, the scheme described in Sec. VI could be used, with the scheme estimating four states instead of three and anticipating some reduced tolerance to sensor noise. With dual-redundant measurements of x_1, x_2, x_3, and x_4, with S_1 and S_2 measuring x_1, S_3 and S_4 measuring x_2, S_5 and S_6 measuring x_3, and S_7 and S_8 measuring x_4, there are several options. S_1–S_3–S_5 could be treated as a unit, S_2–S_4–S_6 could be treated as a unit, and the prefilter could generate estimates of the state vector using (S_1–S_3–S_5), (S_2–S_4–S_6), (S_1–S_3–S_5)–S_7, (S_1–S_3–S_5)–S_8, (S_2–S_4–S_6)–S_7, (S_2–S_4–S_6)–S_8. A comparison of the averaged outputs of sensor pairs, as described in Sec. VI, could be used to identify the failed sensors exactly. Another option is to treat (S_1–S_3) as a unit, (S_2–S_4) as a unit, (S_5–S_6) as a unit, and (S_7–S_8) as a unit and let the prefilter generate estimates of the state vectors using (S_1–S_3), (S_2–S_4), (S_1–S_3)–(S_5–S_6), (S_1–S_3)–(S_7–S_8), (S_2–S_4)–(S_5–S_6), (S_2–S_4)–(S_7–S_8). A comparison of the averaged outputs of sensor pairs will exactly identify the failed sensors. There are other options, but these are two possible ones.

Suppose the plant is higher-order plus multi-input/multi-output like a spacecraft attitude determination system. Now the states of the system would be three Euler angles that describe the spacecraft attitude and the three components of the spacecraft angular velocity in body axes. The inputs to the system would be the three components of the applied torque to the spacecraft acting about its center of mass expressed in body axes. Assume that a star tracker and a sun sensor are used to reconstruct the three Euler angles describing the spacecraft attitude. This star tracker and sun sensor pair could be denoted as sensing system S_1 that measures the three states describing the spacecraft attitude. Assume another star tracker and sun sensor pair is also used to reconstruct the three Euler angles describing the spacecraft attitude. This star tracker and sun sensor pair could be denoted as sensing system S_2. Assume there is a package of three single-axis rate sensors that measures the three components of the spacecraft angular velocity in body axes. Call this package sensing system S_3 that measures the three states describing the spacecraft angular velocity. Assume there is a second package of three single-axis rate sensors that also measures the three components of the spacecraft angular velocity in body axes. Call this package sensing system S_4. Then the new estimation scheme could be configured so that the prefilter estimates the six states of the system using S_1, S_2, S_1–S_3, S_1–S_4, S_2–S_3, S_2–S_4. This is like the second-order plant with dual-redundant sensors, although in this case there are really six states and 10 sensors. A comparison of pairs of sensor outputs as described in Sec. VI could be used to identify failed sensors exactly. This illustrates how the new estimation scheme can be applied to a wide variety of plants, including those that are higher-order plus multi-input/multi-output.

VIII. Conclusions

This paper has presented a new estimation scheme that offers optimal state estimation with failed sensor discrimination and identification. The scheme was successfully applied to second- and third-order plants with dual-redundant measurements of the system states. Strategies for its application to plants of fourth order and higher and plants that are multi-input/multi-output were presented. Thus, the new estimation is applicable to a wide variety of plants.

If this work is extended, it would be desirable to add the capability to identify failed actuators to the scheme. This is an area that is currently being investigated. Testing on a real-world problem is in order. This could be the navigation system for a launch vehicle, a missile, or a deep space probe; an attitude determination system for a free-flying spacecraft; a flight-control system for an aircraft; or the state estimation system in a nuclear power plant. These are just a few of its possible applications.

References

[1] Brown, R. G., and Hwang, P. Y. C., *Introduction to Random Signals and Applied Kalman Filtering, Second Edition*, Wiley, New York, 1992, pp. 181–267.

[2] Willsky, A. S., "A Survey of Design Methods for Failure Detection in Dynamic Systems," *Automatica*, Vol. 12, No. 6, 1976, pp. 601–611.

[3] Montgomery, R. C., and Price, D. B., "Failure Accommodation in Digital Flight Control Systems Accounting for Nonlinear Aircraft Dynamics," *Journal of Aircraft*, Vol. 13, No. 2, 1976, pp. 76–82.

[4] Patton, R. J., Chen, J., and Nielson, S. B., "Model-Based Methods for Fault Diagnosis: Some Guidelines," *Transactions of the Institute of Measurement and Control*, Vol. 17, No. 2, 1995, pp.73–83.

[5] Anderson, G. C., Quinn, D. A., Beals, G. A., Nelson, J. D., and Nurre, G. S., "An Overview of the Hubble Space Telescope Pointing Control System Design and Operation," AIAA Paper 92-4616, Aug. 1992.

[6] Schauwecker, C., Shawger, S., Tung, F., and Nurre, G., "Imaging Pointing Control and Aspect Determination System for the NASA Advanced X-Ray Astrophysics Facility," American Astronautical Society, AAS Paper 97-064, Feb. 1997.

[7] Moore, F. B., and White, J. B., "Application of Redundancy in the Saturn V Guidance and Control System," AIAA Paper 67-553, Aug. 1967.

[8] Gelb, A., *Applied Optimal Estimation*, MIT Press, Cambridge, MA, 1984, pp. 107–119, 291, 292.

[9] Polites, M. E., "Ideal State Reconstructor for Deterministic Digital Control Systems," *International Journal of Control*, Vol. 49, No. 6, 1989, pp. 2001–2011.

[10] Jacquot, R. G., *Modern Digital Control Systems*, Marcel Dekker, New York, 1981, pp. 126, 127.

[11] Meditch, J. S., *Stochastic Optimal Linear Estimation and Control*, McGraw–Hill, New York, 1969, pp. 176–185.

[12] Goodwin, G. C., and Sin, K. S., *Adaptive Filtering, Prediction, and Control*, Prentice–Hall, Englewood Cliffs, NJ, 1984, p. 32.

[13] Greville, T. N., "The Pseudoinverse of a Rectangular or Singular Matrix and Its Application to the Solution of Systems of Linear Equations," *SIAM Review*, Vol. 1, No. 1, 1959, p. 38.