Table 2 Miss distance comparison for systems with and without radome slope compensation

R_m	Method[a]	$A_{31} = 2.0$ s, m	$A_{31} = 3.0$ s, m	$A_{31} = 4.0$ s, m
0.02	X	0.95	1.63	2.21
	I	0.10	0.15	0.21
	II	0.71	1.06	1.53
0.04	X	1.43	2.73	4.39
	I	0.16	0.21	0.33
	II	0.98	1.42	2.28
0.06	X	1.68	4.54	7.59
	I	0.19	0.36	0.61
	II	1.23	2.14	3.57

[a]X = without compensation, I = with compensation in seeker's tracking loop, II = with compensation in guidance loop.

in Table 2 for comparison. It can be seen that the performance of this compensation method is good only for smaller values of radome slope or turning rate time constant. The reason for the degraded performance is that there is additional seeker dynamic lag with the body rate coupling signal of the second method; thus the phase difference between the coupled body rate and the compensating signal is larger than that of the first method, and the uncompensated body rate coupling signal will degrade the performance. However, the advantage of the second method is that it is very easy to be realized and can be applied if the value of radome slope or turning rate time constant is not too large.

Conclusion

This Note presents two methods of radome slope compensation with multiple-model Kalman filters and a semi-Markov Bayesian estimation technique. The semi-Markov transition property among the models of radome slope is very practical and can save computation time and memory size. By weighting those model parameter values with the associated a posteriori probabilities, one can estimate the radome slope and design the compensator in the seeker's tracking loop or in the guidance loop. Various simulations have been given to show that the guidance performances of the system can be improved, and the method of compensation in the seeker's tracking loop is better. However, it is easier to implement the compensator in the guidance loop. Therefore, one should trade off these two methods for practical consideration.

References

[1]Nesline, F. W., and Zarchan, P., "Missile Guidance Design Trade-offs for High Altitude Air Defense," *Journal of Guidance, Control, and Dynamics*, Vol. 6, 1983, pp. 207–212.

[2]Nesline, F. W., and Zarchan, P., "Radome Induced Miss Distance in Aerodynamically Controlled Homing Missiles," *Proceedings of AIAA Guidance and Control Conference*, AIAA, Washington, DC, 1984, pp. 99–115.

[3]Garnell, P., and East, D. J., *Guided Weapon Control Systems*, Pergamon, New York, 1977.

[4]Nesline, F. W., and Zarchan, P., "Wing Size vs Radome Compensation in Aerodynamically Controlled Radar Homing Missiles," *Journal of Guidance, Control, and Dynamics*, Vol. 9, 1986, pp. 645–649.

[5]Nesline, F. W., and Zarchan, P., "Miss Distance Dynamics in Homing Missiles," *Proceedings of AIAA Guidance and Control Conference*, AIAA, Washington, DC, 1984, pp. 84–98.

[6]Stengel, R. F., *Stochastic Optimal Control, Theory and Application*, Wiley, New York, 1986.

[7]Kramer, S. C., and Sorenson, H. W., "Bayesian Parameter Estimation," *IEEE Transactions on Automatic Control*, Vol. AC-33, No. 2, 1988, pp. 217–222.

[8]Moose, R. L., and Wang, P. P., "An Adaptive Estimation with Learning for a Plant Containing Semi-Markov Switching Parameter," *IEEE Transactions on Syst.-Man, Cybern*, Vol. SMC-3, 1973, pp. 277–281.

[9]Moose, R. L., and Wang, P. P., "An Adaptive State Estimation Solution to the Maneuvering Target Problem," *Transactions on Automatic Control*, Vol. AC-20, 1975, pp. 359–362.

[10]Chang, C. B., and Athans, M., "State Estimation for Discrete Systems with Switching Parameters," *Transactions on Aerospace Electronic Systems*, Vol. AES-14, 1978, pp. 418–427.

[11]Howard, R. A., "System Analysis of Semi-Markov Process," *IEEE Transactions on Military Electronics*, Vol. Mil-8, 1964, pp. 114–124.

[12]Yueh, W. R., "Adaptive Estimation Scheme for Radome Error Calibration," *Proceedings of IEEE 22nd CDC Conference*, Inst. of Electrical and Electronics Engineers, 1983, pp. 546–551.

[13]Yueh, W. R., and Lin, C. F., "Bank-to-Turn Guidance Performance, Analysis with In-Flight Radome Error Compensation," *Proceedings of AIAA Guidance and Control Conference*, AIAA, Washington, DC, 1984, pp. 715–722.

[14]Nesline, F. W., "Missile Guidance for Low-Altitude Air Defense," *Journal of Guidance and Control*, Vol. 2, 1979, pp. 283–289.

Sensor Fault Detection and Diagnosis for a T700 Turboshaft Engine

Jonathan Litt*
NASA Lewis Research Center, Cleveland, Ohio 44135
Mehmet Kurtkaya†
Oyak Renault Automobile Company, Bursa, Turkey
and
Ahmet Duyar‡
Florida Atlantic University, Boca Raton, Florida 33431

Introduction

A PROPOSED intelligent control system (ICS) contains a sensor fault detection, isolation, and accommodation scheme.[1] If the differences between the estimated and sensed system outputs exceed some threshold values, fault detection logic will initiate a parameter estimation algorithm which determines the type and magnitude of the failure. This is achieved through the use of fault parameters, variables defined to convert the model from that of the unimpaired system to that of an impaired one. Once the fault is isolated, the control system will accommodate it, if possible.[2] This Note discusses the technique used for detecting, isolating, and identifying the fault.

The test bed for this research is the T700 turboshaft engine. In the simplified model used here there is one input, fuel flow W_f, and four measured variables, gas generator speed N_g, interturbine gas temperature T_{45}, interturbine gas pressure P_{45}, and power turbine torque output Q_{PT}. Should any of the sensors fail, the engine would appear to be malfunctioning and, if the faulty measurement were fed back through the control system, the engine would operate off the design point.

Sensor Fault Detection and Isolation

The sensor fault detection and isolation scheme is developed using a fault model. Initially a simplified linear perturbation model was developed which switches between several point models for full envelope coverage.[3] The simplified model at an operating point is of the standard form

$$\left.\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ y(k) &= Cx(k) \end{aligned}\right\} \tag{1}$$

The variables in these point models are all normalized with their zero values corresponding to the operating point. It is assumed that the (A, B, C) realization of the system is in α-canonical form,[4] which gives the model certain properties desirable for fault detection.

Received Feb. 7, 1994; revision received July 14, 1994; accepted for publication. Aug. 23, 1994. This paper is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

*Aerospace Engineer, Vehicle Propulsion Directorate, Army Research Laboratory, Mail Stop 77-1, 21000 Brookpark Road.

†Engineer, Research and Development.

‡Professor, Mechanical Engineering Department.

The sensor fault model is built on top of Eq. (1) in the form

$$\left.\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ y_s(k) &= F_s Cx(k) + f_{s0} \end{aligned}\right\} \quad (2)$$

where $y_s(k)$ is the set of normalized sensor readings at time k. In the unimpaired case, F_s is the identity matrix and f_{s0} is the zero vector, thus reducing Eq. (2) to Eq. (1). The diagonal matrix F_s accounts for changes in sensor gain whereas a nonzero f_{s0} represents bias errors.

A nonlinear simulation of the T700 turboshaft engine and load[5] was used to represent the engine. The time step used is 0.060 s.

Past controller outputs as well as the sensor readings are fed through the fault detection model described by Eq. (2), and the resulting estimate of the set of sensed variables is compared to those received from the simulation at the current time step. The differences, or residuals, are checked against threshold values which, when exceeded, activate the fault detection scheme.

The fault detection scheme contains models appropriate for each type of failure, i.e., actuator, component, and sensor. Just as the sensor fault parameters are associated with the C matrix as shown in Eq. (2), the actuator fault parameters are associated with the B matrix and the component fault parameters are associated with the A matrix.[1] When the residual exceeds the activation threshold, an on-line parameter estimation scheme is initiated which calculates the fault parameters, F_s and f_{s0} in the sensor case.

It is important to run the parameter estimation scheme using data from the impaired system so that the results are not skewed by prefailure information. Therefore, the parameter estimation module begins collecting data to use only after the activation threshold is exceeded. A recursive least squares technique with a forgetting factor of 0.98 is used to compute the value of the fault parameters. As long as the fault occurs suddenly, it appears as a step change in the residuals. Thus, for the short time it takes to perform the identification, the measurement signals are rich enough to allow for accurate estimation. The models of the actuator, component, and sensor failures are developed in parallel, and the three sets of fault parameters are computed simultaneously in the ICS.

At each time step, the resulting fault parameters are passed to the hypothesis testing module which determines from the identified parameters which fault occurred. It assumes that only one failure occurs at a time. The result is checked by confirming that the selected model using the identified fault parameters produces a smaller sum of the squared errors (residuals) over a finite time than do the other estimated models using their fault parameters. In this case a moving window of five data points was used for summing the squares of the residuals before comparing the models, resulting in a delay of at least 0.3 s (5 samples × 0.060 s/sample) before the fault is isolated and its magnitude estimated. This data length was experimentally determined to produce reliable results in relatively few samples.

Results

Testing was performed at the 96% power level. There was no noise present in the system. For the first case, a sensor bias of 1.8% of nominal was introduced to the nonlinear simulation's N_g sensor's output at about 0.0 s. As soon as the bias was added, the corrupted signal disrupted all of the other output variables. The fault parameters corresponding to the multiplicative sensor gain F_s converged immediately to their correct values of 1.0 as determined through recursive least squares, indicating no sensor gain failure. Figure 1 shows the bias estimates f_{s0}. The bias estimate for the N_g sensor converged to approximately 0.018 whereas the other bias values stayed near zero. The slight error in the P_{45} bias is attributed to modeling error in the linear point model.

For the second case, the original unity gain of the N_g sensor was scaled by a factor of 0.1 at about 0.0 s. Figure 2 shows the multiplicative gain fault parameters as determined by the recursive least squares technique from past data. It takes significantly longer to converge than the previous example did but eventually the parameter corresponding to the N_g sensor's gain converges to about 0.1 whereas the other three hover near 1.0. Figure 3 displays the estimated values of the bias parameters, which, after some initial hunting, converge to approximately 0.0.

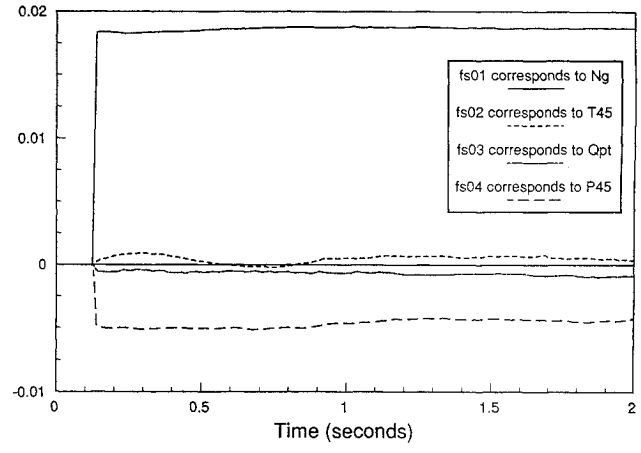

Fig. 1 Bias fault parameters for a 1.8% bias error in N_g sensor.

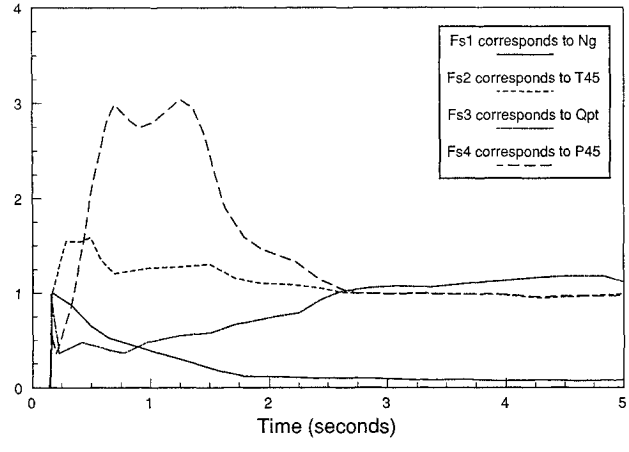

Fig. 2 Multiplicative fault parameters for a 0.1 multiplicative error in N_g sensor.

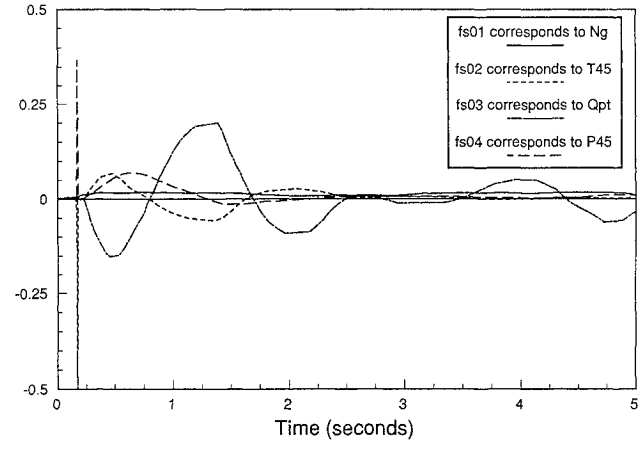

Fig. 3 Bias fault parameters for a 0.1 multiplicative error in N_g sensor.

Conclusions

A scheme to detect, isolate, identify and estimate the type and magnitude of sensor faults in a T700 turboshaft engine has been developed and demonstrated. The injection of a fault caused the residual to exceed the threshold value and trigger the identification scheme. Once running, the scheme produced accurate results reasonably quickly. The identified fault parameters can be used in the

state equation by an ICS to eliminate the bias or cancel the incorrect gain, thereby accommodating the sensor faults and allowing the closed-loop system to run as if unimpaired.

Several issues remain to be investigated once noise is included in the system: the effect on threshold values against which residuals are compared, the ability of an appropriate parameter identification technique to provide bias-free estimates, and the minimum number of samples included in the moving window for convergence of the estimates.

Acknowledgment

This work was funded by the Army Research Laboratory's Vehicle Propulsion Directorate under Grant NAG3-1198.

References

[1]Duyar, A., Eldem, V., Merrill, W., and Guo, T.-H., "Fault Detection and Diagnosis in Propulsion Systems: A Fault Parameter Estimation Approach," *Journal of Guidance, Control, and Dynamics*, Vol. 17, No. 1, 1994, pp. 104–108.

[2]Litt, J. S., "An Expert System to Perform On-Line Controller Restructuring for Abrupt Model Changes," American Helicopter Society Rotary Wing Propulsion Specialists' Meeting, Paper 19, Williamsburg, VA, Nov. 13–15, 1990.

[3]Duyar, A., Gu, Z., and Litt, J. S., "A Simplified Dynamic Model of the T700 Turboshaft Engine," NASA TM 105805, AVSCOM TR 92-C-024, June 1992.

[4]Eldem, V., and Duyar, A., "Parametrization of Multivariable Systems Using Output Injections: Alpha Canonical Forms," *Automatica*, Vol. 29, No. 4, 1993, pp. 1127–1131.

[5]Ballin, M. G., "A High Fidelity Real-Time Simulation of a Small Turboshaft Engine," NASA TM 100991, July 1988.

On-Line Robust Stabilizer

R. Balan* and D. Aur†

University "Politehnica" of Bucharest, 77206 Bucharest, Romania

I. Introduction

THIS Note presents an adaptive robust discrete solution for a stabilization problem. As an application of this solution, a stability augmentation system (SAS) for an aircraft is presented.

The general scheme of this system is given in Fig. 1. Throughout this Note two hypotheses are assumed: 1) full information about the state of the system ($y_k = x_k$) and 2) that the identification block gives the best fit of the linearized time-varying nonlinear system (i.e., $x_{k+1} = A_k x_k + B_k u_k$) (see, e.g., Ref. 1).

The sample time of the control loop is much less than that of the identification and optimization loop. This allows on-line identification and optimization procedures.

A convenient criterion to be optimized is sought such that the control u_k will be given by a linear state feedback: $u_k = F_k x_k$. This criterion will include two terms: one involving the performance requirements and the other involving the stability robustness:

$$C_{PR} = \lambda C_P + C_R$$

where λ is a weighting parameter.

II. Optimization Problem

The feedback matrix F_k will be given by a Riccati equation, but instead of a classical discrete algebraic Riccati equation (DARE),

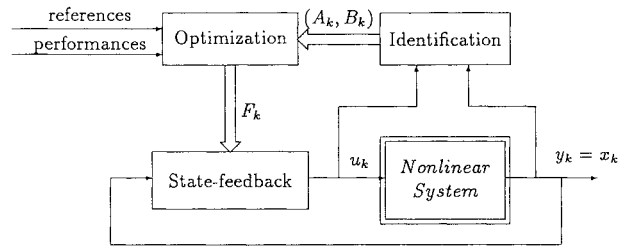

Fig. 1 General scheme.

a modified DARE will be used such that the eigenvalues of the stable closed-loop matrix are to be within some disk included in the unit disk (according to Ref. 2 this is called the D-pole assignment problem). Let us consider the disk $D(\alpha, r)$ of center α and radius r in the complex plane such that α is a real number and $|\alpha| + r \leq 1$ [i.e., $D(\alpha, r) \subset D(0, 1)$]. In order to state the criterion, the following DARE is considered:

$$\tilde{A}_k^T K_k \tilde{A}_k - K_k - \tilde{A}_k^T K_k B_k \left(R + B_k^T K_k B_k\right)^{-1} \times B_k^T K_k \tilde{A}_k + Q = 0 \tag{1}$$

where $\tilde{A}_k = (A_k - \alpha I)/r$ is the modified A matrix and $Q, R > 0$ are positive matrices related to the quadratic cost:

$$I = \sum_{j \geq k} \left(x_j^T Q_k x_j + u_j^T R u_j\right) \tag{2}$$

with

$$Q_k = Q + \tilde{A}_k^T K_k \tilde{A}_k - A_k^T K_k A_k + A_k^T K_k B_k \left(R + B_k^T K_k B_k\right)^{-1} \times B_k^T K_k A_k - \tilde{A}_k^T K_k B_k \left(R + B_k^T K_k B_k\right)^{-1} B_k^T K_k \tilde{A}_k \tag{3}$$

and the linear dynamics $x_{j+1} = A_k x_j + B_k u_j$, $j \geq k$.

It is known that there exists a unique stabilizable solution $K_k = K_k^T > 0$ of Eq. (1).[3] Let us set

$$F_k = -r\left(R + B_k^T K_k B_k\right)^{-1} B_k^T K_k \tilde{A}_k \tag{4}$$

and

$$A_{s,k} = A_k + B_k F_k \tag{5}$$

Since $\Lambda(\tilde{A}_k + B_k F_k / r) \subset D(0, 1)$, one can obtain

$$\Lambda(A_{s,k}) \subset D(\alpha, r) \tag{6}$$

where $\Lambda(\cdot)$ denotes the eigenvalues set of the matrix.

Now the criterion to be minimized can be stated as follows:

Performance part:

$$C_P = x_k^T K_k x_k \tag{7}$$

Stability robustness part:

$$C_R = x_k^T A_{s,k}^T A_{s,k} x_k = \|A_{s,k} x_k\|^2 \tag{8}$$

The first part of the criterion is related to the quadratic cost (2) of the dynamics and the matrices Q and R are chosen to fulfil the performance requirements.

The second part gives an ∞-norm bound along the trajectory: the goal is to minimize not the H_∞-norm of A_s (which is $\max_{\|x\|=1} \|A_s x\|$) but just the $\|A_s x\|$ along the trajectory; since it is not known a priori the trajectory, in an on-line solution only $\|A_s x\|$ with x taken to be the actual state ($x = x_k$) is to be minimized. The form of C_R is suggested by the constrained stability measure introduced in Ref. 4, taking for G the real trajectory of our system and also $P = I$ in the formula (17) of the cited paper.

Received Jan. 31, 1994; revision received June 18, 1994; accepted for publication Oct. 17, 1994.

*Ph.D. Student, Department of Automatic Control and Computers.

†Ph.D. Student, Department of Aeronautics.