criteria for directional instability phenomena.[6] The derivation of the D_{lat} criterion in the present Note, however, clearly shows that terms involving gravity do have an important role to play in the dynamic directional instability phenomenon of wing rock onset. In fact, when the effect of gravity is neglected by allowing the factor (g/V_0) to go to zero in Eq. (9), the approximate criterion for wing rock onset reduces to $C_{n_{\beta\text{dyn}}} = 0$, which, as seen before, merely signifies static instability.

V. Conclusions

An approximate analytical criterion for dynamic directional instability, called D_{lat}, has been derived that provides a useful and simpler alternative to the exact Routh discriminant condition for wing rock onset. The D_{lat} criterion reveals, for the first time, an explicit link between wing rock onset, a dynamic instability phenomenon, and the $C_{n_{\beta\text{dyn}}}$ parameter.

Acknowledgments

Partial support for this work by way of a grant-in-aid from the Aeronautics Research and Development Board (Aerodynamics Panel), government of India, is gratefully acknowledged. The authors are also thankful to the anonymous reviewers for their helpful comments.

References

[1]Abzug, M. J., and Larrabee, E. E., *Airplane Stability and Control—A History of the Technologies that Made Aviation Possible*, Cambridge Univ. Press, Cambridge, England, U.K., 1997, pp. 131–139.

[2]Ross, A. J., "Investigation of Nonlinear Motion Experienced on a Slender Wing Research Aircraft," *Journal of Aircraft*, Vol. 9, No. 9, 1972, pp. 625–631.

[3]Duncan, W. J., *Control and Stability of Aircraft*, Cambridge Univ. Press, Cambridge, England, U.K., 1952, pp. 117–125.

[4]Moul, M. T., and Paulson, J. W., "Dynamic Lateral Behavior of High Peformance Aircraft," NACA RM L58E16, Aug. 1958.

[5]Calico, R. A., Jr., "A New Look at $C_{n_{\beta\text{dyn}}}$," *Journal of Aircraft*, Vol. 16, No. 12, 1979, pp. 893–896.

[6]Lutze, F. H., Durham, W. C., and Mason, W. H., "Unified Development of Lateral-Directional Departure Criteria," *Journal of Guidance, Control, and Dynamics*, Vol. 19, No. 2, 1996, pp. 489–493.

[7]Liebst, B. S., and Nolan, R. C., "Method for the Prediction of the Onset of Wing Rock," *Journal of Aircraft*, Vol. 31, No. 6, 1994, pp. 1419–1421.

[8]Livneh, R., "Improved Literal Approximation for Lateral–Directional Dynamics of Rigid Aircraft," *Journal of Guidance, Control, and Dynamics*, Vol. 18, No. 4, 1995, pp. 925–927.

[9]Phillips, W. F., "Improved Closed Form Approximation for Dutch Roll," *Journal of Aircraft*, Vol. 37, No. 3, 2000, pp. 484–490.

[10]Ananthkrishnan, N., and Unnikrishnan, S., "Literal Approximations to Aircraft Dynamic Modes," *Journal of Guidance, Control, and Dynamics*, Vol. 24, No. 6, 2001, pp. 1196–1203.

[11]Newell, F. D., "Ground Simulator Evaluations of Coupled Roll-Spiral Mode Effects on Aircraft Handling Qualities," U.S. Air Force Flight Dynamics Lab., Rept. AFFDL-TR-65-39, 1965.

[12]Roskam, J., *Airplane Flight Dynamics and Automatic Flight Controls, Part I*, DARcorporation, Lawrence, KS, 1995, p. 348.

[13]Shah, P., Unnikrishnan, S., and Ananthkrishnan, N., "Analytical Prediction of Wing Rock Onset for Aircraft with Lightly Damped Lateral Dynamics," AIAA Paper 2003-5304, Aug. 2003.

[14]Van der Meer, J. C., *Hamiltonian Hopf Bifurcation*, Lecture Notes in Mathematics, Vol. 1160, Springer-Verlag, Berlin, 1985, p. 15.

[15]Ananthkrishnan, N., and Sudhakar, K., "Characterization of Periodic Motions in Aircraft Lateral Dynamics," *Journal of Guidance, Control, and Dynamics*, Vol. 19, No. 3, 1996, pp. 680–685.

[16]Fan, Y., Lutze, F. H., and Cliff, E. M., "Time-Optimal Lateral Maneuvers of an Aircraft," *Journal of Guidance, Control, and Dynamics*, Vol. 18, No. 5, 1995, pp. 1106–1112.

[17]Ananthkrishnan, N., and Sinha, N. K., "Level Flight Trim and Stability Analysis Using Extended Bifurcation and Continuation Procedure," *Journal of Guidance, Control, and Dynamics*, Vol. 24, No. 6, 2001, pp. 1225–1228.

Learning-Based Sensor Validation Scheme Within Flight-Control Laws

M. L. Fravolini*
Perugia University, 06125 Perugia, Italy
and
G. Campa,† M. R. Napolitano,‡ and M. Perhinschi¶
West Virginia University,
Morgantown, West Virginia 26506-6106

I. Introduction

RESEARCH on analytical redundancy (AR) and model-based sensor fault detection identification and accomodation (SFDIA) has shown that AR[1] can provide fault-tolerance capabilities for aircraft where physical redundancy in the onboard sensors is not a cost-effective solution. Although effective schemes for linear-time-invariant systems[2] are already established, the extension of these schemes to nonlinear systems still presents substantial research challenges.[1] In Ref. 3 three of the authors proposed a sensor fault detection isolation accomodation (SFDIA) scheme based on neural-network (NN) approximators. In this effort this scheme has been improved through a new fault-identification logic. Failures were simulated in closed-loop conditions, and a detailed analysis was performed to determine the smallest detectable and identifiable faults along with the relative detection and isolation delay. A new hybrid configuration for the neural approximators was also introduced.

II. Analytical Redundancy-Based SFDIA Scheme for the Longitudinal Aircraft Dynamics

The main variables involved in the aircraft longitudinal dynamics are angle of attack α (rad), pitch rate q (rad/s), normal acceleration $a_z(g)$, elevator deflection δ_e (rad), airspeed V (m/s), and altitude $H(m)$. The AR between these signals is embedded in the equations of motion of the aircraft through the normal force equation:

$$ma_z = \rho(H)V^2SC_z(\alpha, \delta_e, V, q/V)/2 \tag{1}$$

The AR between the variables in Eq. (1) was used to implement the nonlinear input/output models required in the SFDIA scheme. In particular, the following estimation models were developed:

$$\hat{\alpha}(k) = f_\alpha\left(V(k), H(k), a_z(k)/V^2(k), q(k)/V(k), \delta_e(k)\right) \tag{2}$$

$$\hat{q}(k) = f_q\left(V(k), H(k), V(k), a_z(k)/V^2(k), \alpha(k), \delta_e(k)\right) \tag{3}$$

$$\hat{a}_z(k) = f_{a_z}\left(V(k), H(k), V(k), q(k)/V(k), \alpha(k), \delta_e(k)\right) \tag{4}$$

$$\hat{\delta}_e(k) = f_{\delta_e}\left(V(k), H(k), a_z(k)/V^2(k), \alpha(k), q(k)/V(k)\right) \tag{5}$$

In the preceding equations, k indicates the current time instant. The use of q/V and a_z/V^2 in lieu of a_z and q as inputs allows the keeping

Received 12 November 2001; revision received 28 January 2003; accepted for publication 6 October 2003.

*Visiting Research Assistant Professor, Department of Electronic and Information Engineering, Via G. Duranti 93.

†Research Assistant Professor, Department of Mechanical and Aerospace Engineering.

‡Professor, Department of Mechanical and Aerospace Engineering. Member AIAA.

¶Associate Professor, Department of Mechanical and Aerospace Engineers.

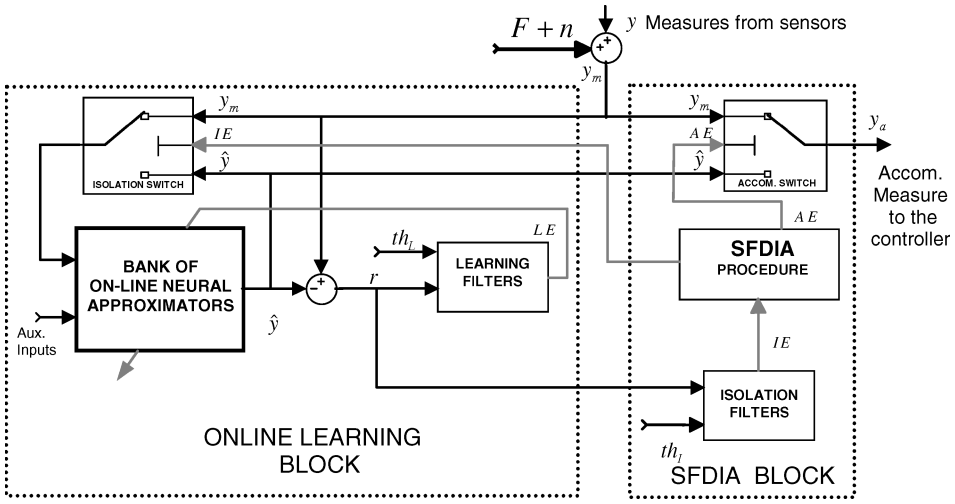

Fig. 1 NNs learning enabling and SFDIA architecture.

of the functions in Eqs. (2–5) as smooth as possible. The scheme is based on the evaluation of the following residuals signals $r_i(k)$:

$$r_i(k) = y_i(k) - \hat{y}_i(k) + n_i(k) + F_i(k - k_f), \qquad i = 1 \cdots 4 \quad (6)$$

where $y_i(k)$ is the ith measurement, $\hat{y}_i(k)$ is the estimation, $n_i(k)$ is the measurement noise, and $F_i(k - k_f)$ is the additive failure modeling function, where k_f is the fault beginning instant. A linear ramp-like shape is selected to model the failure using

$$F(k - k_f) = \begin{cases} 0 & k < k_f \\ A \cdot (k - k_f)/T_R & k_f \le k < k_f + T_R \\ A & k \ge k_f + T_R \end{cases} \quad (7)$$

where T_R is the duration of the ramp and A is the final value. Different combinations of T_R and A allow the modeling of both hard or soft failures. It is assumed that a fault can occur only on the α, q, a_z, and δ_e sensors. In Ref. 3 the authors used radical basis function resource allocating networks (RBF-RANs)[4] for the online approximation of functions similar to $f_q(\cdot)$, $f_\alpha(\cdot)$, $f_{a_z}(\cdot)$, $f_{\delta e}(\cdot)$. In this work, the approximation architecture was improved by the addition of a linear in the parameters (ADALINE) neural network:

$$\hat{y}(x(k), p, \theta(k)) = \hat{y}_{\text{lin}}(x(k), p) + \hat{y}_{\text{ran}}(x(k), \theta(k))$$

$$= \sum_{i}^{N} p_i \cdot x_i(k) + \sum_{i=1}^{N_{nn}} w_i(k) \cdot e^{\left[\frac{|x(k) - \mu_i(k)|^2}{2\sigma_i^2(k)}\right]} \quad (8)$$

where $x(k)$ and $\theta(k)$ are respectively the RAN inputs and weights vector. In this particular study the N parameters p_i of the ADALINE are not time varying but have been evaluated offline by applying the least-squares (LS) method to a set of nominal training data. The online learning is performed by the RAN algorithm that allocates neurons only where nonlinearities are dominant. A block diagram of the scheme, which can be classified as a generalized observer method,[1] is shown in Fig. 1. In the scheme $y_m(k)$ are measured variables, affected by measurement noise $n(k)$ and by potential sensor failures $F(k)$, $\hat{y}_i(k)$ are the estimates provided by the NNs, and $y_a(k)$ are the fault-free signals, also called accommodated signals. The core of the block is the bank of NNs providing the estimates $\hat{y}_i(k)$, used to generate the residual $r(k) = y_m(k) - y_i(k)$; $r(k)$ goes through a bank of learning filters, each one defined by

$$r_{\text{Li}}(k) = b_{\text{Li}} \cdot r_{\text{Li}}(k - 1) + (1 - b_{\text{Li}}) \cdot r_i(k), \qquad b_{\text{Li}} = e^{-a_{\text{Li}}^T} \quad (9)$$

where a_{Li} is the continuous time constant of the filter. The filtered residuals are used to generate the Boolean NNs learning (LE) vector: when the absolute value of a component of $r_{\text{Li}}(k)$ exceeds the corresponding threshold th_{Li}, the scheme deduces that an abnormal condition is occurring and LE_i is set to zero to prevent the NN learning from measurements corrupted by a possible failure.

The Boolean signal Isolation Enable (*IE*) is generated by a bank of isolation filters $r_I(k)$ that is identical to the one in Eq. (9), except for the different thresholds th_I and time constants a_I. Upon a failure occurrence, one or more components of the Boolean vector *IE* change, and thus the scheme detects that a sensor fault is occurring. After a detection a dedicated residual evaluation procedure is called with the double objective of providing identification of the failed sensor and providing an accommodation by substitution of the failed sensor with its estimation $\hat{y}_i(k)$. This isolation and accommodation procedure is based on the consideration that in the case of a failure on the ith sensor two situations can occur:

1) The fault is correctly attributed to the ith sensor; therefore, the FDIA procedure commands the isolation switch to substitute the measurement $y_{\text{mi}}(k)$ with its estimate $\hat{y}_i(k)$, then $\hat{y}_i(k)$ is sent to all of the NNs instead of the faulty measure. In this case, the estimate $\hat{y}_i(k)$ is accurate because it is evaluated using fault-free measurements from the remaining "healthy" sensors. Thus, because $\hat{y}_i(k) \cong y_i(k)$ the corresponding residual is proportional to the fault amplitude. Therefore, its value tends to remain high confirming the persistency of the fault $F_i(k - k_f)$ on the ith sensors; in this case, the other residuals tend to decrease because after the commutation $\hat{y}_i(k) \to y_i(k)$ their outputs return to be evaluated with fault-free estimates $\hat{y}_i(k)$.

2) The fault is incorrectly attributed to a jth healthy sensor. Therefore, the isolation switch substitutes the correct measurement $y_{\text{mj}}(k)$ with its estimate $\hat{y}_j(k)$ and provides it to all of the NNs using that signal. In this case, the estimate $\hat{y}_j(k)$ would not be accurate beause it is evaluated using the failure-corrupted measurement $y_i(k)$. Because of this incorrect isolation, after a short transient T_o (T_o is a design parameter) other residuals will tend to exceed their thresholds.

The FDIA logic is such that at the end of the observation period T_o if more than one component of r_I is still above the threshold (as happens in case 2) the next potentially failed sensor is selected as $j = \arg\max(M_{Il}/th_{Il}) \in \{\text{suspect sensors} - i\}$.

III. Simulation Results

For the offline identification of the nonlinear relations in Eqs. (2–5), a flight was simulated at closed-loop conditions for 4000 s within the entire flight envelope. After the simulation, first the four ADALINE-NNs were trained using the LS method; next, the RAN-NNs were trained repeating the RAN learning algorithm 100 times. The thresholds th_L and th_I were selected so that a threshold violation never occurs for the fault-free validation flight. Specifically, the vector th_I was defined as the 120% of Mr_I, and th_L was defined as the 80% of Mr_L, where $Mr_L = \text{Max}(r_L(k)) = (64 \cdot 10^{-2}, 87 \cdot 10^{-2}, 21 \cdot 10^{-2}, 43 \cdot 10^{-2})$, $Mr_I = Mr_L$, $a_L = a_I = 0.1$, and $T = 0.05$ s. The observation period T_o was set at 1.5 s.

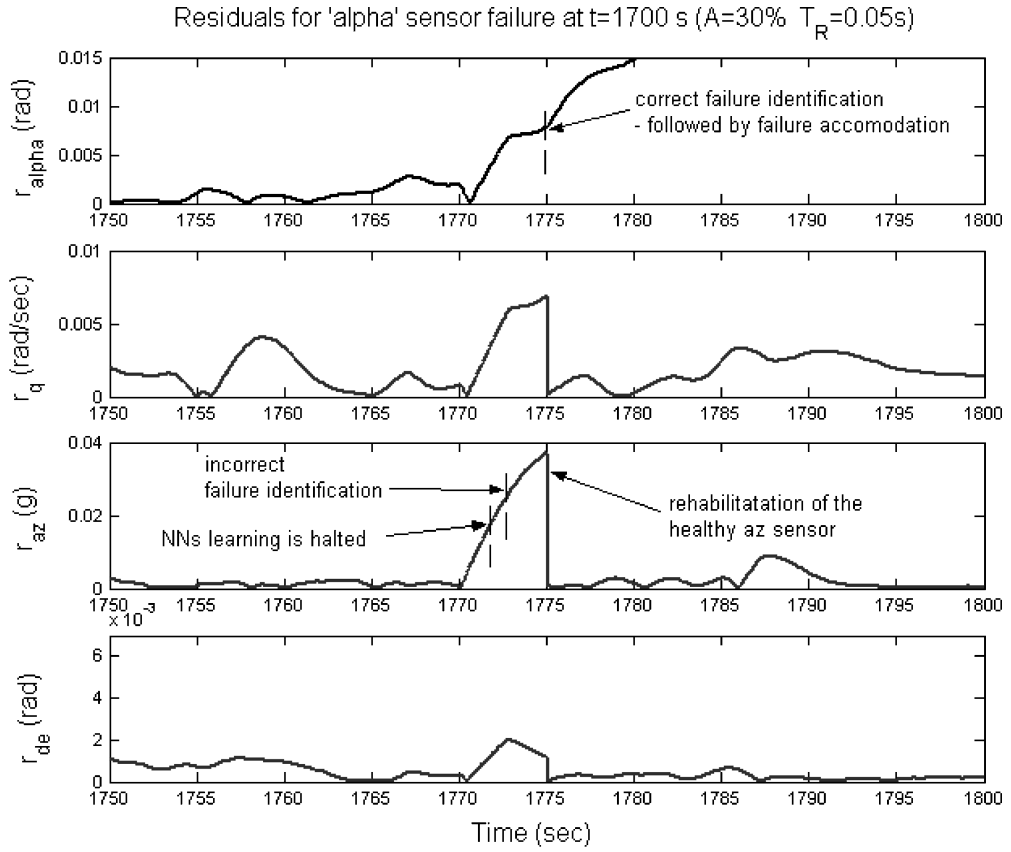

Fig. 2 Magnitude of the residuals and logic signals for α sensor failure.

Using the pretrained NNs and the settings from the offline phase, the SFDIA scheme was implemented at closed-loop conditions on a different (validation) flight. Two types of fault were used:

1) Type 1 fault is the sudden bias ($A = (100, 80, 50, 40, 30, 20, 10, 5)\%$ of the maximum value on the signals $y_m(k)$ evaluated on the validation flight, with fast transient: $T_R = 0.05$ s).

2) Type 2 fault is the drifting bias ($A = (100, 80, 50, 40, 30, 20, 10, 5)\%$ of the maximum value on the signals $y_m(k)$ evaluated on the validation flight, with slow transient $T_R = 10$ s).

Eighty different tests were conducted; in each test the whole system starts at nominal conditions, and all of the NNs are enabled to learn. At $t = t_f$ (in this study $t_f = K \cdot 50$ s $K = 1, \ldots, 80$) a failure is injected; at the end of each test the following performance indices were evaluated: 1) $\mathrm{DE}_{\%}$, percentage of correct detections; 2) $\mathrm{IS}_{\%}$, percentage of correct isolations; 3) T_{LE}, the mean of the time delay (respect to t_f) at which the online learning of the NNs is stopped (switching to 0 of at least one component of the vector LE); 4) T_{IE}, the mean of the time delay (respect to t_f) at which a fault is detected (switching to 0 of at least one component of the vector IE); 5) T_{AE}, the mean of the time delay (respect to t_f) at which the SFDIA procedure commands the final accommodation; and 6) σ_{ee}, standard deviation of the estimation error $\hat{y}_i(k) - y_i(k)$, which measures the accuracy of the approximation after the accommodation (until the end of flight).

Table 1 reports the performance associated with the minimum values of the faults amplitudes that gave a percentage of fault isolation ($\mathrm{IS}_{\%}$) greater than 85%, in case of single fault on the four sensors. Overall, the values of T_{IE} and T_{AE} show good detection/isolation capabilities, and the low values of the index σ_{ee} show the effectiveness of the accomodation at postfailure condition. Figure 2 shows the evolution of the residual vector $\boldsymbol{r}_I(k) = \boldsymbol{r}_L(k) = [r_{I\alpha}(k), r_{\mathrm{Iq}}(k), r_{Ia_z}(k), r_{I\delta e}(k)]$ for a type 1 failure on the α sensor ($A = 30\%$, $T_R = 0.05$ s, $t_f = 1700$ s). At $t = 1772.25$ s the online learning on the NNs is halted because $r_{\mathrm{Laz}}(k)$ exceeds its threshold th_L. Succesively, at $t = 1773.45$ s a fault on the a_z sensor is temporarly (and incorrectely) isolated, the switch $\hat{a}_{\mathrm{zm}}(k) \rightarrow \hat{a}_z(k)$ is enabled, and the observation period T_o starts. At the end of T_o (at $t = 1774.95$ s), through the $r_{L\alpha}(k)$ signal the SFDIA logic attributes the failure to the α sensor and commands the correct accomodation $\hat{\alpha}_{\mathrm{m}}(k) \rightarrow \hat{\alpha}(k)$. At this point, all of the other residuals decrease because of the correct isolation—except the faulty one—and no futher threshold violations are observed.

Table 1 Performance for failure types 1 and 2 on sensors α, q, a_z, and δ_E

Sensor	F.Ampl.%	$\mathrm{DE}_{\%}$	$\mathrm{IS}_{\%}$	T_{LE}, s	T_{IE}, s	T_{AE}, s	σ_{ee}
$\alpha(k)$	type 15	100	87.0	2.55	3.62	4.12	$310\ \mathrm{e}^{-5}$
$q(k)$	type 15	97.4	92.2	4.10	7.49	8.44	$361\ \mathrm{e}^{-5}$
$a_z(k)$	type 10	100	97.4	1.82	2.48	2.51	$138\ \mathrm{e}^{-4}$
$\delta_e(k)$	type 30	100	85.4	1.25	1.64	10.7	$182\ \mathrm{e}^{-5}$
$\alpha(k)$	type 20	100	94.8	6.60	7.77	10.8	$315\ \mathrm{e}^{-5}$
$q(k)$	type 15	97.4	90.9	9.23	12.8	13.4	$365\ \mathrm{e}^{-5}$
$a_z(k)$	type 10	100	90.9	6.67	7.75	7.77	$139\ \mathrm{e}^{-4}$
$\delta_e(k)$	type 30	100	85.8	4.65	5.70	17.8	$180\ \mathrm{e}^{-5}$

IV. Conclusions

This paper describes several improvements for a previously introduced sensor fault detection identification and accomodation scheme. A new hybrid architecture for the approximators has been proposed and tested using a combination of ADALINE- neural networks (NNs) and RAN-NNs. A novel supervisory scheme based on a detailed analysis of the residuals has been introduced and successfully tested through simulations at closed-loop conditions. The proposed innovations considerably enhance the reliability of the scheme.

References

[1]Chen, J., and, Patton, R. J., *Robust Model-Based Fault Diagnosis for Dynamic Systems*, Kluwer Academic, Norwell, MA, 1999.

[2]Frank, P. M., and Ding, X., "Survey of Robust Residual Generation and Evaluation Methods in Observer Based Fault Detection Systems," *Journal of Process Control*, Vol. 7, No. 6, 1997, pp. 403–424.

[3]Campa, G., Fravolini, M. L., Napolitano, M. R., Del Gobbo, D., Seanor, B., Yu, G., and Gururajan, S., "On-Line Learning Neural Networks for Sensor Validation of a B777 Flying Model," *International Journal of Robust and Nonlinear Control*, Vol. 12, No. 11, 2002, pp. 987–1007.

[4]Lu, Y., Sundararajan, N., and Saratchandran, P., "Analysis of Minimal Radial Basis Function Network Algorithm for Real-Time Identification of Non-Linear Dynamic Systems," *IEE Proceedings on Control Theory and Application*, Vol. 4, No. 147, 2000, pp. 476–484.

Application of Pseudospectral Methods for Receding Horizon Control

Paul Williams*
Royal Melbourne Institute of Technology, Melbourne, Victoria 3001, Australia

Introduction

THE design of controllers for nonlinear systems is not always easy, and it is often proposed to segment trajectory optimization problems into offline and online control tasks, particularly for complex systems.[1] The offline phase is devoted to solving the full nonlinear optimal control problem, whereas the online phase usually involves employing a neighboring optimal feedback control strategy based on the linearized dynamics. Because the optimal trajectory is time varying, the neighboring feedback controller traditionally relies on solving the time-varying two-point boundary value problem by a backward sweep of the Riccati equation or by transition matrices.[2] This is often both time consuming and numerically unstable. Furthermore, there is no guarantee that perturbations around the reference trajectory will be small, and so applying the linearized equations may not be valid in some cases. To overcome these difficulties Ohtsuka and Fujii[3] proposed a receding-horizon control strategy for nonlinear systems based on the stabilized continuation method. Ohtsuka[4] has extended the method to time-varying systems. This method relies on the explicit integration of the states and costates derived from the calculus of variations and requires fine tuning the stabilization parameters to obtain satisfactory convergence. Yan et al.[5] present a method for solving linear quadratic optimal control problems by transforming the linear time-varying equations into a set of discrete linear algebraic equations using the Legendre pseudospectral method. Yan et al.[6] use the approach of Ref. 5 to generate the inner feedback loop of a two-degree-of-freedom control system. Their method is based on a neighboring optimal control strategy to minimize deviations from the optimal trajectory. Lu[1] has proposed a receding-horizon strategy for precision entry guidance based on an Euler–Simpson approximation of the quadratic programming (QP) problem. Lu[1] employs a Simpson approximation for the integral cost and Euler approximations for the state derivatives. In this way, Lu solves the QP problem analytically to obtain approximate control laws. Whereas the methods proposed by Yan et al.[5,6] and Lu[1] do not require any explicit integration, they are still based on the assumption that deviations from the reference trajectory are within the linear approximation.

Received 10 September 2003; revision received 9 November 2003; accepted for publication 18 November 2003.

*Ph.D. Candidate, School of Aerospace, Mechanical, and Manufacturing Engineering, 15 Ellindale Avenue, McKinnon; tethers@hotmail.com. Student Member AIAA.

In this Note, a method for solving nonlinear receding-horizon control problems by pseudospectral approximations is presented. The proposed solution method approximates the nonlinear receding-horizon control problem with successive linear approximations obtained via the method of quasi linearization. Each linear problem is solved by discretizing the two-point boundary value problem obtained from the calculus of variations using a Jacobi pseudospectral method. This allows the linear problem to be solved using matrix algebra. The approach is very similar to that used by Yan et al.,[5] except that a quasi linearization algorithm is first applied to the nonlinear problem. Another important difference between the proposed method and that of Yan et al.[5] is the way that the boundary conditions are incorporated. Yan et al.[5] use a matrix partitioning approach to obtain feedback control laws from an overdetermined set of equations. In this Note, however, the boundary conditions are applied directly in a similar manner to that done when solving partial differential equations by pseudospectral methods.[7] The method is applied to the difficult problem of retrieving a subsatellite using a flexible tether.

Theory

Consider the nonlinear receding horizon control problem of finding the control input $\boldsymbol{u}(t)$ that minimizes the performance index

$$J = \frac{1}{2}[\boldsymbol{M}_f\boldsymbol{x} - \boldsymbol{\psi}]^T\boldsymbol{S}_f[\boldsymbol{M}_f\boldsymbol{x} - \boldsymbol{\psi}]|_{t+T} + \frac{1}{2}\int_t^{t+T} \times \left\{\begin{bmatrix}(\boldsymbol{x}-\boldsymbol{x}_d)^T & (\boldsymbol{u}-\boldsymbol{u}_d)^T\end{bmatrix}\begin{bmatrix}\boldsymbol{Q} & \boldsymbol{N}\\ \boldsymbol{N}^T & \boldsymbol{R}\end{bmatrix}\begin{bmatrix}(\boldsymbol{x}-\boldsymbol{x}_d)\\ (\boldsymbol{u}-\boldsymbol{u}_d)\end{bmatrix}\right\} \mathrm{d}\tau \quad (1)$$

subject to the state equations and initial conditions

$$\dot{\boldsymbol{x}} = f[\boldsymbol{x}(\tau), \boldsymbol{u}(\tau), \tau], \qquad \boldsymbol{x}(\tau = t) = \boldsymbol{x}(t) \quad (2)$$

where $\boldsymbol{x}(\tau) \in R^n$ is the state vector, $\boldsymbol{x}_d(\tau) \in R^n$ is the desired state vector, $\boldsymbol{u}(\tau) \in R^m$ is the vector of control inputs, $\boldsymbol{u}_d(\tau) \in R^m$ is the desired control input, $t \in R$ is the time, $\tau \in [t, t+T] \in R$ is the time variable used to predict the future system state, $\boldsymbol{\psi} \in R^p$, $p \le n$, is the desired vector of values for the linear combination of desired final states $[\boldsymbol{M}_f\boldsymbol{x}(t+T)]$, $\boldsymbol{Q}$ is a positive semidefinite matrix, $\boldsymbol{N}$ is a given matrix, $\boldsymbol{R}$ is a positive definite matrix, $\boldsymbol{S}_f$ is the positive semidefinite terminal weighting matrix, $\boldsymbol{M}_f$ is a given matrix, and T is the length of the future horizon. Note that there is a distinction between the state and control variables used for predicting the future state, parameterized by τ, and the actual state and control variables, parameterized by t. Once the problem defined by Eqs. (1) and (2) is solved, the actual control input is applied at the current time by $\boldsymbol{u}(t) = \boldsymbol{u}(\tau = t)$.

To solve this problem, we employ the method of quasi linearization in a manner similar to that by Jaddu[8] and Xu and Agrawal.[9] In this method, the performance index is expanded up to second order and the system equations are expanded to first order around nominal trajectories. The solution to the original problem is obtained by solving the sequence of resulting linear optimal control problems. This differs from other methods of quasi linearization that first apply the necessary conditions for optimality and then linearization, where an update parameter is employed to aid convergence.[10] Application of the method of quasi linearization to Eqs. (1) and (2) results in the following sequence of linear optimal control problems: To minimize

$$\tilde{J} = \frac{1}{2}[\boldsymbol{M}_f\tilde{\boldsymbol{x}} - \boldsymbol{\psi}]^T\boldsymbol{S}_f[\boldsymbol{M}_f\tilde{\boldsymbol{x}} - \boldsymbol{\psi}]|_{t+T} + \frac{1}{2}\int_t^{t+T} \times \left\{\begin{bmatrix}(\tilde{\boldsymbol{x}}-\boldsymbol{x}_d)^T & (\tilde{\boldsymbol{u}}-\boldsymbol{u}_d)^T\end{bmatrix}\begin{bmatrix}\boldsymbol{Q} & \boldsymbol{N}\\ \boldsymbol{N}^T & \boldsymbol{R}\end{bmatrix}\begin{bmatrix}(\tilde{\boldsymbol{x}}-\boldsymbol{x}_d)\\ (\tilde{\boldsymbol{u}}-\boldsymbol{u}_d)\end{bmatrix}\right\} \mathrm{d}\tau \quad (3)$$

subject to

$$\dot{\tilde{\boldsymbol{x}}} = \underline{\boldsymbol{A}}(\tau)\tilde{\boldsymbol{x}} + \underline{\boldsymbol{B}}(\tau)\tilde{\boldsymbol{u}} + \underline{\boldsymbol{w}}(\tau), \qquad \tilde{\boldsymbol{x}}(\tau = t) = \tilde{\boldsymbol{x}}(t) \quad (4)$$