the second layer a log sigmoid, and the third layer a linear network. They have nine neurons at each layer. The training method used in this study is Levenberg–Marquardt backpropagation method (see Ref. 4).

To demonstrate the optimality of the solutions resulting from the neural networks, we obtained optimal solutions to this minimum time problem using a shooting method[1] for one set of initial conditions. Flight Mach number histories from a shooting method, labeled two-point boundary value problem (TPBVP), and the neural networks are presented in Fig. 1. They are almost coincident showing that the neural solutions obtained with the cascade of controllers are (near) optimal. The initial Mach number for this selective example is 0.8. The associated angle-of-attack (control) histories are presented in Fig. 2. For most of the flight they are the same for both neural networks and the shooting method. The differences can be accounted for by the fact that the shooting method uses several more control steps than the 37 steps with neural networks. Although these plots establish the (near) optimality of the neurosolutions, the real advantage in using the adaptive critic approach is demonstrated in Fig. 3. For each trajectory with initial Mach number varying from 0.6 to 0.8, the final Mach number is 0.8. That is, the same cascade of neurocontollers is used to generate optimal control for an envelope of initial conditions. For all of the trajectories, the thrust profile was assumed to be constant. We carried out further numerical experiments to test the robustness of the controllers. We removed six controllers in the mid-flight-path region (which meant that some of the controls are held constant for longer periods) and plotted the missile trajectory. Even though the trajectory is less optimal, the lesser network configuration still delivers the missile to the exact final Mach number of 0.8.

Another advantage/flexibility of using these network controllers is that we can vary the initial flight-path angles for various Mach numbers. In these cases, all that is required to generate the optimal control to meet the final condition is to start from the network indexed with the initial flight-path angle and proceed to the next controllers as the flight-path angles changes. This is true by Bellman's principle of optimality (see Ref. 1), which states that, on an optimal path, trajectory from any intermediate stage to final stage is optimal for the given cost function.

VI. Conclusions

An adaptive critic-based neural network solution for a free final time problem associated with agile missile control has been developed. To our knowledge, there has been no other tool (other than dynamic programming) that provides such solutions. Also, the computational effort associated with the adaptive critic-based solutions is not prohibitive. This computational technique is fairly general and applicable to a wide variety of problems.

Acknowledgments

This research was supported by U.S. Air Force Grant F08630-96-1-0001 and National Science Foundation Grant ECS-9634127.

References

[1] Bryson, A. E., and Ho, Y., *Applied Optimal Control*, Hemisphere, New York, 1975, pp. 128–211.

[2] Stengel, R. F., *Optimal Control and Estimation*, Dover, New York, 1994, Chap. 3.

[3] Hunt, K. J., "Neural Networks for Controller Systems, A Survey," *Automatica*, Vol. 28, No. 6, 1992, pp. 1083–1112.

[4] White, D. A., and Sofge, D., *Handbook of Intelligent Control*, Van Nostrand Reinhold, New York, 1992, Chaps. 3, 5, 8, 12, 13.

[5] Balakrishnan, S. N., and Biega, V., "Adaptive Critic Based Neural Networks for Aircraft Optimal Control," *Journal of Guidance, Control, and Dynamics*, Vol. 19, No. 4, 1996, pp. 893–898.

[6] Anderson, C. W., "Learning to Control an Inverted Pendulum Using Neural Networks," *IEEE Control Systems Magazine*, Vol. 9, No. 3, 1989, pp. 31–37.

[7] Barto, A. G., Sutton, R. S., and Anderson, C. W., "Neuronlike Adaptive Elements That Can Solve Difficult Learning Control Problems," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. SMC-13, 1983, pp. 834–846.

[8] Werbos, P., "Neurocontrol and Supervised Learning; An Overview and Evaluation," *Handbook of Intelligent Control*, Van Nostrand Reinhold, New York, 1992, Chap. 3.

Jump Markovian-Based Control of Wing Deployment for an Uncrewed Air Vehicle

A. Stoica*
University "Polytechnica" of Bucharest,
Bucharest RO 77206, Romania
and
I. Yaesh†
Israel Military Industries, 47100
Ramat-Hasharon, Israel

I. Introduction

AIR-LAUNCHED uncrewed air vehicles (UAVs) are often released with their wings folded to achieve clear and safe separation. The wings are deployed only when the UAV is required to begin a significant glide slope maneuver (see URL: http// vectorsite.tripod.com/avbomb9.html and URL: http//www.imi-israel.com). The ensuing abrupt change in aerodynamic forces and moments cause significant disturbances and result in a jump of the aerodynamic coefficients, thus leading to a transient in the angle of attack, which should be minimized to avoid loss of stability.

Note that the roll control problem also presents a challenge due to possible flow asymmetries that occur during the wing deployment transition period, resulting in large roll angle transients.

In the present Note, the focus is on the pitch control problem, where it is assumed that wing deployment is fast enough to assure that the system may be described by either wings-folded or wings-unfolded dynamics. This assumption stems from that speedy deployment mechanisms, able to complete their operation within 0.05−0.3 s, are relatively low cost and simple to implement, for example, by a pneumatic piston. Slow deployment requires an appropriate servomechanism to achieve a smoothly controlled wing unfolding process. To avoid system complexity and the higher cost of such a servomechanism, a rapid wing deployment mechanism is preferred. Because the 0.05−0.3 s wing deployment time is of the same order of magnitude as the aerodynamic short period, no intermediate wing positions need to be considered during the design process. Both force and moment coefficients are assumed, for design purposes, to undergo a jump at the instant of wing deployment, with the jump value depending on the instantaneous angle of attack. The angle of attack minimizing the lift during the jump phase differs from the one minimizing the pitch moment at the jump instant, and a compromise angle of attack of 0 deg is, therefore, chosen as the angle of attack for wing opening.

The UAV under consideration is equipped with a pitch rate sensor, a normal force accelerometer, and an angle-of-attack sensor. It is also equipped with a potentiometer indicating wing position. The zero-angle-of-attack design goal may also be interpreted as a 0-g requirement, thereby motivating the design of a 0-g commanded acceleration loop.

The aim of this Note is to present the design of a pitch acceleration loop whose task it is to minimize the possibly harmful effects of wing deployment. One possible approach involves embedding the plant into a convex polytope,[1] whose vertices contain the dynamic descriptions of the folded and unfolded wing aerodynamics,

Received 9 April 2001; revision received 13 September 2001; accepted for publication 21 November 2001.

*Professor, Faculty of Aerospace Engineering, Str. Splaiul Independentei No. 313; stoica@aeronet.propulsion.pub.ro.

†Head of Control Department, Advanced Systems Division, P.O.B. 1044/77; mlyaesh@isdnmail.co.il.

respectively, and designing a constant gain controller for said system. The resulting controller is said to be quadratically stabilizing and guarantees disturbance attenuation for any convex combination of the two vertices of the polytope.

Such an approach, however, involves an overdesign, that is, is conservative, and may result in an overly large disturbance attenuation factor γ. Note that stability of the closed-loop system is assured by this method, in spite of the instantaneous jump in the system's parameters. To reduce the overdesign, the two vertices may be dealt with separately by designing a different controller for each vertex, thus resulting in a lower γ. In such a situation, neither performance nor stability is guaranteed for convex combinations of the two vertices. In addition, smooth system behavior resulting from the transient effect of the system's parameter jumps cannot be guaranteed. One possible way of circumventing the shortcomings of the preceding two methods is to combine both of them via the Markovian jump system control methodology (see Ref. 2, and, particularly, Ref. 3 and the references therein). In this approach, the system parameters jump from a wings-folded to a wings-open state with well-defined transition probabilities, thus allowing for the existence of distinct controllers for each system, while explicitly taking into account the transients resulting from the jump. In the present Note, the preceding paradigm for the design of an acceleration loop and the testing of its performance are explored. In the next section, the wing deployment problem of a UAV is posed. In Sec. III, the necessary and sufficient conditions for solvability of the problem are presented. In Sec. IV, three controller designs are described, and design comparisons are made. Conclusions are then made in Sec. V.

Throughout the Note the superscript T stands for matrix transposition, $\mathcal{R}^n$ denotes the n-dimensional Euclidean space, $\mathcal{R}^{n\times m}$ is the set of all $n\times m$ real matrices, and the notation $P>0$ (respectively, $P\geq 0$) for $P\in\mathcal{R}^{n\times n}$ means that P is symmetric and positive definite (respectively, semidefinite). Throughout, $(\Omega, \mathcal{F}, \mathcal{P})$ is a given probability space; the argument $\theta\in\Omega$ will be suppressed. Expectation is denoted by $E\{\cdot\}$, and conditional expectation of x on the event $\theta(t)=i$ is denoted by $E[x\mid\theta(t)=i]$.

II. Wing Deployment Problem

A. General Description

In a typical UAV normal acceleration control loop, the servo, modeled as a first-order system, receives elevon commands δ_{e_c} and accordingly moves the elevon surface to δ_e. The elevon commands are supplied by the controller, whose inputs are the vertical component $V_w(t)$ of the true air speed vector $\boldsymbol{V}(t)$, the angular pitch rate $q(t)$, the normal acceleration $a_z(t)$, and the wing's position. Until the wings unfold, the controller also receives a zero acceleration command $a_{z_c}=0$, thereby reflecting the desired near zero value of the angle of attack $\alpha=V_w/U_0$, until the instant at which the wings begin to unfold. Note that U_0 is the magnitude of the true air speed vector. The aerodynamic model of the UAV reacts to elevon surface movements δ_e and to disturbances in $\dot{V}_w$ and $\dot{q}$. This combination of lift force and pitch moment disturbances results from the changes in the lift and pitch moment coefficients C_L and C_M, respectively, between the closed-wing and open-wing aerodynamic configurations of the UAV. Unfortunately, no angle of attack in which the jumps in both coefficients are simultaneously minimal could be found. Therefore, the required zero angle of attack at the wing deployment instant is the result of a compromise in trying to keep both jumps below a reasonable level. Immediately following wing deployment, the acceleration command should ideally jump from a 0-g to a 1-g level (acceleration command for straight and level flight), but in practice, this jump may be modified to be a ramp function.

B. Plant Model

To illustrate the effectiveness of the proposed control method, consider the transfer function T_{a_z,δ_e}, which relates the control surface position δ_e to the normal acceleration a_z measured at l_x, a location ahead of the center of gravity ($l_x=4.92$ ft). This transfer function is based on a short-period approximation[4] of an air vehicle and is given by

Table 1 Linearization results

Coefficient	Units	Folded	Open
Z_w	1/s	−1.0767100e−01	−4.6275600e−01
Z_q+U_0	ft/s	7.1853400e+02	7.1718900e+02
Z_{δ_e}	1/s	−3.1367200e+01	−1.6713900e+01
M_{δ_e}	1/s	−1.9531600e+01	−1.1363800e+01
Z_q	ft/s	2.8870465e+00	2.8209838e+00
M_w	1/ft	−2.1881300e−02	−3.3291000e−02
M_q	1/s	−7.2090600e−01	−7.5222700e−01

$$\dot{\boldsymbol{x}}_a = A_a\boldsymbol{x}_a + B_a\delta_e + G_a\boldsymbol{d} = \frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix} V_w \\ q \end{bmatrix}$$

$$= \begin{bmatrix} Z_w & Z_q+U_0 \\ M_w & M_q \end{bmatrix}\begin{bmatrix} V_w \\ q \end{bmatrix} + \begin{bmatrix} Z_{\delta_e} \\ M_{\delta_e} \end{bmatrix}\delta_e + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\boldsymbol{d}$$

$$a_z = Z_wV_w + Z_qq + Z_{\delta_e}\delta_e - l_x\dot{q} = C_a\begin{bmatrix} V_w \\ q \end{bmatrix} + D_a\delta_e$$

$$C_a = [Z_w - l_xM_w \quad Z_q - l_xM_q], \qquad D_a = Z_{\delta_e} - l_xM_{\delta_e}$$

where V_w is the vertical component of the true air speed vector in body-axis coordinates and q is the pitch rate. The disturbance vector $\boldsymbol{d}=[\dot{V}_{w\,\mathrm{dist}}, \dot{q}_{\mathrm{dist}}]^T$ reflects the changes in the lift force and pitching moment, respectively, between the wings-folded and wings-unfolded conditions. The dimensional aerodynamic coefficients are given by

$$Z_w = -\frac{\rho SU_0}{m}C_{L_\alpha}, \qquad Z_q = -\frac{\rho SU_0c}{4m}C_{L_q}$$

$$M_w = \frac{\rho SU_0c}{2I_{yy}}C_{M_\alpha}, \qquad M_q = \frac{\rho SU_0c^2}{4I_{yy}}C_{M_q}$$

$$Z_{\delta_e} = \frac{\rho SU_0^2}{2m}C_{L_{\delta_e}}, \qquad M_{\delta_e} = \frac{\rho SU_0^2}{2I_{yy}}C_{M_{\delta e}}$$

The nondimensional coefficients (for example, see Ref. 4) C_{L_α}, C_{L_q}, C_{M_α}, C_{M_q}, $C_{L_{\delta_e}}$, and $C_{M_{\delta_e}}$, attain different values for the wings-folded and wings-open configurations. The true air speed is U_0, the air density is ρ, the mass is m, and the moment of inertia about the y-body axis is I_{yy}. The reference chord and area are c and S, respectively.

At a specific operating point, where the two cases reflect the wings-folded and wings-open values of the nondimensional coefficients, the dimensional coefficients for the nominal case were computed by linearization around trim conditions, that is, zero state derivatives, at a zero angle of attack, of the six-degrees-of-freedom equations (for example, see Ref. 4). The results shown in Table 1 were obtained.

C. Plant Augmentation

To enable controller design, the plant needs to be augmented to include the elevon servomodel and dynamic weightings. The elevon servo is modeled by $\dot{\delta}_e = -(\delta_e - \delta_{e_c})/\tau$, where $\tau = 1/30$ s.

That the acceleration control loop should force the measured normal acceleration to track the acceleration command can easily be incorporated into the design process by introducing an integral action on the acceleration tracking error via $\dot{\xi} = a_z - a_{z_c}$.

The state and disturbance vectors are then augmented to become $\boldsymbol{x} = [V_w \;\; q \;\; \delta_e \;\; \xi]^T$ and $w = [a_{z_c} \;\; \dot{V}_{w\mathrm{dist}} \;\; \dot{q}_{\mathrm{dist}}]^T$, respectively. The output signal to be minimized is chosen as $\boldsymbol{z} = [\beta\xi \;\; \rho\delta_{e_c}]^T$. The augmented plant then becomes, $\dot{\boldsymbol{x}} = A\boldsymbol{x} + B\delta_{e_c} + G\boldsymbol{w}$, $\boldsymbol{z} = L\boldsymbol{x} + D\delta_{e_c}$ where

$$A = \begin{bmatrix} A_a & B_a & 0_{2,1} \\ 0_{1,2} & -30 & 0 \\ C_a & D_a & 0 \end{bmatrix}, \qquad B = \begin{bmatrix} 0_{2,1} \\ 30 \\ 0 \end{bmatrix}$$

$$G = \begin{bmatrix} 0_{2,1} & I_2 \\ 0 & 0_{1,2} \\ -1 & 0_{1,2} \end{bmatrix}, \qquad L = \begin{bmatrix} 0_{1,2} & 0 & \beta \\ 0 & 0 & 0 \end{bmatrix}, \qquad D = \begin{bmatrix} 0 \\ \rho \end{bmatrix}$$

The parameter $\beta = 20$ is the required bandwidth of the closed-loop system in tracking the acceleration command, and $\rho = 100$ is the control signal (elevon command) weighting. An elevon command signal $\delta_{e_c} = K[\omega(t)]x$ is sought such that $J < 0$, where J, defined by Eq. (2), (see Sec. III), implies that the induced norm of the operator relating w to z should be less than γ.

III. Necessary and Sufficient Solvability Conditions

Consider the following system:

$$\dot{\boldsymbol{x}}(t) = A[\theta(t)]\boldsymbol{x}(t) + G[\theta(t)]\boldsymbol{w}(t) + B[\theta(t)]\boldsymbol{u}(t), \qquad x_0 = 0$$
$$z(t) = L[\theta(t)]\boldsymbol{x}(t) + D[\theta(t)]\boldsymbol{u}(t) \tag{1}$$

where $\boldsymbol{x}(t) \in \mathcal{R}^n$ are the system states that may include the states of a dynamic weighting function used to shape the closed-loop frequency response, $\boldsymbol{w}(t) \in \mathcal{R}^q$ is the exogenous disturbance signal, and $z(t) \in \mathcal{R}^m$ is the combination of the input signals $\boldsymbol{u}(t)$ and states to be minimized, with possible dynamic weighting.

The matrices A, G, B, D, and L are piecewise constant matrices of appropriate dimensions whose entries are dependent on the wing state [$\theta(t) = 1 \Rightarrow$ wings folded or $\theta(t) = 2 \Rightarrow$ wings open]. For design purposes, it is assumed that $\theta(t)$, $t \geq 0$, is a right continuous homogeneous Markov chain on $\mathcal{D} = \{1, 2\}$ with a probability transition matrix

$$P(t) = e^{Qt}, \qquad Q = [q_{ij}], \qquad q_{ii} < 0$$
$$\sum_{j=1}^{2} q_{ij} = 0, \qquad i = 1, 2$$

Given the initial condition $\theta(0) = 1$, that is, the wings are initially folded, at each time instant t, the wings may maintain their current state, that is, remain folded or opened if currently so, or jump, that is, unfold if currently folded. The transitions between the two wing states, $i \in \mathcal{D}$, occur in accordance with current wing status and the transition probabilities derived from the transition rate matrix Q. Appropriate values for the entries of Q should reflect that, once the wings have been deployed, they may never return to their folded state. Q describing the preceding situation consists of

$$Q = \begin{bmatrix} -1 & 1 \\ 0.01 & -0.01 \end{bmatrix}$$

corresponding to the transition probabilities as a function of time as shown in Fig. 1a, where the solid line depicts $P_{1,1}$ and the dash-dotted line represents $P_{1,2}$. It is also assumed that $D^T[D, L] = [R, 0]$, where $R > 0$. Note that $Q(2, 2) = -Q(2, 1) = -0.01$ was assumed rather than a zero value, reflecting the very small probability of the wings refolding without violating the requirement of $q_{ii} < 0$. Consider the following cost function:

$$J = E\int_0^\infty [z^T(t)z(t) - \gamma^2\boldsymbol{w}^T(t)\boldsymbol{w}(t)]\,\mathrm{d}t \tag{2}$$

Piecewise constant gain state feedback controllers of the type $\boldsymbol{u}(t) = K[\theta(t)]\boldsymbol{x}$ are sought, where $K[\theta(t)] = K_i$, $i = 1, 2, \theta(t) = i$. When the latter is substituted into Eq. (1), the closed-loop system becomes

$$\dot{\boldsymbol{x}}(t) = \{A[\theta(t)] + BK[\theta(t)]\}\boldsymbol{x}(t) + G[\theta(t)]\boldsymbol{w}(t), \qquad x_0 = 0$$
$$z(t) = \{L[\theta(t)] + D[\theta(t)]K[\theta(t)]\}\boldsymbol{x}(t) \tag{3}$$

Denoting by $A_i = A[\theta(t)]$, $B_i = B[\theta(t)]$, $C_i = C[\theta(t)]$, $G_i = G[\theta(t)]$, $L_i = L[\theta(t)]$, and $D_i = D[\theta(t)]$, where $\theta(t) = i$, $i = 1, 2$, and writing the bounded real lemma (for example, Ref. 2, Corollary 7.11) for the resulting system (3) leads to

$$\begin{bmatrix} (A_i + B_iK_i)^TX_i + X_i(A_i + B_iK_i) + \displaystyle\sum_{j=1}^{2} q_{ij}X_i + (L_i + D_iK_i)^T(L_i + D_iK_i) & X_iG_i \\ G_i^TX_i & -\gamma^2 I \end{bmatrix} < 0, \qquad i = 1, 2 \tag{4}$$

Note that the more general output feedback problem has been solved in Ref. 3. For the sake of completeness, the derivation of the simpler state feedback problem solution for two jump states, under the simplifying orthogonality assumption $D^T[D, L] = [R, 0]$, is now presented. Inequality (4) can be written in the equivalent form

$$Z_i + P_i^TK_iQ_i + Q_i^TK_i^TP_i < 0 \tag{5}$$

where, for $i \neq j$,

$$Z_i \triangleq \begin{bmatrix} \tilde{A}_i^TX_i + X_i\tilde{A}_i + q_{ij}X_j + L_i^TL_i & X_iG_i & 0 \\ G_i^TX_i & -\gamma^2 I & 0 \\ 0 & 0 & -I \end{bmatrix}$$

$$P_i^T \triangleq \begin{bmatrix} X_iB_i \\ 0 \\ D_i \end{bmatrix}, \qquad Q_i \triangleq [I \quad 0 \quad 0]$$

and where $\tilde{A}_i = A_i + (q_{ii}/2)I$. Condition (5) is then equivalent to (see Ref. 5, p. 22) $W_{P_i}^TZ_iW_{P_i} < 0$ and $W_{Q_i}^TZ_iW_{Q_i} < 0$, where W_{P_i} and W_{Q_i} are bases of the null spaces of P_i and Q_i, respectively, and are composed of

$$W_{P_i} = \begin{bmatrix} 0 & X_i^{-1}W_{1i} \\ I & 0 \\ 0 & W_{2i} \end{bmatrix}, \qquad W_{Q_i} = \begin{bmatrix} 0 & 0 \\ I & 0 \\ 0 & I \end{bmatrix}$$

where

$$\begin{bmatrix} W_{1i} \\ W_{2i} \end{bmatrix}$$

is a basis of the null space of $[B_i^T \;\; D_i^T]$. The condition $W_{Q_i}^TZ_iW_{Q_i} < 0$ is then automatically fulfilled. Furthermore, $W_{P_i}^TZ_iW_{P_i} < 0$ leads to

$$\begin{bmatrix} -\gamma^2 I & G_i^TW_{1i} \\ W_{1i}^TG_i & W_{1i}^TX_i^{-1}\big(\tilde{A}_i^TX_i + X_i\tilde{A}_i + q_{ij}X_j + L_i^TL_i\big)X_i^{-1}W_{1i} - W_{2i}^TW_{2i} \end{bmatrix} < 0$$

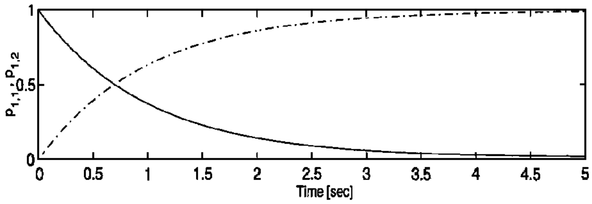

Fig. 1a Transition probability.

When $S_i \overset{\Delta}{=} X_i^{-1}$ is defined, it follows that the preceding inequality is equivalent to the following linear matrix inequality (LMI) for i, $j = 1, 2, i \neq j$,

$$\begin{bmatrix} W_{1i}^T\left(\tilde{A}_i S_i + S_i \tilde{A}_i^T + \gamma^{-2} G_i G_i^T\right)W_{1i} - W_{2i}^T W_{2i} & W_{1i}^T S_i L_i^T & \sqrt{q_{ij}}\, W_{1i}^T S_i \\ L_i S_i W_{1i} & -I & 0 \\ \sqrt{q_{ij}}\, S_i W_{1i} & 0 & -S_j \end{bmatrix} < 0 \tag{6}$$

The following result is therefore obtained. Also see Ref. 3 for the situation where the solution lies within the set of dynamic nth-order controllers.

Theorem. The cost function of Eq. (2) is negative if and only if there exist symmetric positive definite matrices S_i, $i = 1, 2$ satisfying Eq. (6). In such a case, the controller gains K_i, $i = 1, 2$, are obtained from LMI (5).

Note that the controller gains jump concurrently with the system's parameter jumps, as does the worst-case disturbance strategy. In a way, the resulting controller is, therefore, gain scheduled. The jump in the system's parameters are, however, reflected in the cost function of Eq. (2), and the possibly resulting transients are, thus, minimized. In the forthcoming discussion, this controller will be referred to as the Markovian jump controller (MJC). If one chooses to design a single controller, the quadratically stabilizing controller of Ref. 5, which is conservative because it does not use the information available about the system's parameters, that is, $i = 1$ or 2, is the result. In the sequel, the controller of Ref. 5 will be referred to as the robust controller (RC).

IV. Controller Design

The controller was designed in three different ways, the RC method of Ref. 5, the design of a separate H_∞ (SDH) controller at each vertex, and the MJC. The RC controller achieved a minimum closed-loop disturbance attenuation factor of $\gamma = 33.43$, with corresponding gains of $K = [10.57\ \ -425.6\ \ -180.7\ \ -305.7]$. The gains at each vertex resulting from the SDH method, which ensure minimum disturbance attenuation factors of 18.1 and 12.9, are $K_1 = [0.0040\ \ -0.0825\ \ -0.7510\ \ -0.4253]$ and $K_2 = [0.1212\ \ 1.2540\ \ -1.7674\ \ -1.6579]$, respectively. Finally, the MJC was designed by applying the preceding theorem, where the LMI there was solved using the tools of Ref. 6. The minimum closed-loop disturbance attenuation factor for the MJC approach is considerably less than that of RC and is $\gamma = 20.0$. The gains for the wings-folded and wings-unfolded conditions are $K_1 = [0.0290\ \ -2.7269\ \ -1.1120\ \ -1.5065]$ and $K_2 = [0.0110\ \ 0.7722\ \ -0.4793\ \ -0.2112]$, respectively. Note the marked differences between the gains of the folded and deployed wings as in the SDH approach. The different gains for the corresponding different plant parameters, together with that quadratic stability is not required for MJC, may explain the improvement in γ with respect to RC.

A. Comparison Issue

It may be claimed that the comparison between the disturbance attenuation levels obtained using the deterministic (RC and SDH) and the stochastic (MJC) frameworks is meaningless. One possible way of addressing such a claim is by comparing the solutions on an equal footing. Two such comparison approaches were, therefore, chosen, the first being completely deterministic and the second entirely stochastic in nature. In the first approach, the H_∞ norm of the closed loops for $i = 1$ and 2 for all three solutions were computed. The norms so obtained were 32.9 and 22.5 for RC, 18.1 and 12.9 for SDH, and 18.3 and 15.7 for MJC, which are remarkably lower than those of the polytopic controller (RC) and surprisingly close to those of SDH. Therefore, in the deterministic sense, MJC is a reasonable compromise between the RC and the SDH methods.

In the second approach, the performance levels were compared on the basis of the stochastic Markovian jump theoretical bounded real lemma of Ref. 2. These performance levels were readily computed by substituting the gains K_i, $i = 1, 2$, into Eq. (3) to obtain the closed-loop systems for all three methods (with $K_1 = K_2 = K$ for RC), and applying the bounded real lemma of Ref. 2. The stochastic disturbance attenuation level for RC was 32.4, and for SDH it was 76.7. This result is not surprising because the method does not take the parameter jumps into account. The corresponding MJC level was 20.0 and is consistent with the design value and markedly below the other designs.

It is common practice to choose a control approach, which, although not entirely suited to the problem from a theoretical viewpoint, nevertheless provides the best of all of the solution alternatives. The most practical comparison method is based on time responses, and the results of this approach now follow.

B. Simulations

The simulations were performed on the model in Sec. II.C, with a step change in lift of 18 ft/s^2 at the instant of wing opening, and a corresponding step change of 0.56 rad/s in pitch rate as a result of the step change in the pitching moment. The wings were assumed to be deployed at 5 s. The initial conditions are $V_{w_0} = \alpha_0 \pi / 180$, $q_0 = 0$, $\delta_{e_0} = 0$, and $\xi_0 = 0$, where $\alpha_0 = 2$ deg is the initial angle of attack. The acceleration command is zero throughout the folded-wing phase of the flight and jumps to -1 $g = 32.174$ ft/s^2 (pitch up) at the wing deployment instant. In fact, it was assumed that the wings are completely deployed within 0.1 s, during which both plant and controller data are linearly interpolated between the extreme values corresponding to $i = 1$ and 2. Similarly, the lift and pitch rate moment disturbances, as well as the change in the acceleration command, are linearly varied within the 0.1-s period.

The results are shown in Figs. 1b and 1c, where the angle of attack and elevon angle, respectively, are shown for the three control design methods (dashed lines for RC, dotted lines for MJC, and dash–dotted lines for SDH) at times between 4.5 and 7 s. From the angle-of-attack overshoot point of view, similar performance was obtained for all three methods, whereas MJC uses considerably less control effort than both RC and SDH. For completeness, different wing deployment durations in the range of 0.05–0.3 were tested.

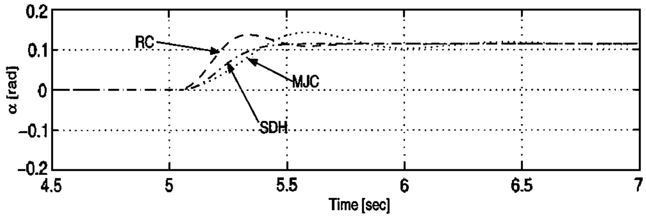

Fig. 1b Angle of attack.

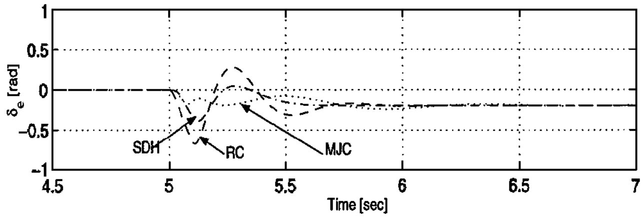

Fig. 1c Elevon angle.

The superiority of MJC is maintained for wing-opening durations of $0.05 - 0.15$, although for wing-deployment periods larger than 0.2, the advantage of MJC over SDH is diminished. Because of uncertainties in the wing deployment mechanism and because larger deployment periods may require more complex mechanisms, durations less than 0.2 s should not be ignored. This implies that MJC appears to be useful for the described application.

V. Conclusions

The synthesis problem of wing deployment of a UAV was considered via the jump Markovian systems approach. The change in the vehicle's dynamics due to wing deployment was given a probabilistic interpretation, where the probability of deploying the wings from a folded position increases exponentially with time. However, once deployed, the wings remain open and, therefore, have a zero probability of switching back to a folded position. This description may even be closer to reality than the more intuitive deterministic one whereby the instant of wing deployment is selected according to a set of a priori defined conditions on the angle attack and its derivative, for example, small absolute value of the angle of attack combined with a small absolute derivative of the angle of attack. Even in the deterministic case, the first moment during which the described conditions are satisfied may depend on random parameters such as the angle of attack of the host vehicle at release, its velocity, etc., giving rise, in a very natural way, to a probabilistic interpretation of the switching time. For the situation where wing deployment timing is defined a priori, the transition probabilities may be considered to be design parameters tuned to maximizing the closed-loop bandwidth, minimizing control effort, etc.

The results achieved by the Markovian jump systems approach are quite encouraging. For very short-duration wing deployment, the Markov-jump-theory-based disturbance attenuation factor is markedly less than those obtained with the more common approaches of simply treating each wing state separately or requiring quadratic stability. The latter uses a single gain matrix to control the plant during its two different phases and, therefore, does not utilize the information about the parameter jumps, thus resulting in poor performance with a large control effort. Separately treating the closed- and open-wing systems does utilize the jump information, but does not account for the transient, thus leading to good disturbance attenuation, but at the cost of large controls. The Markovian jump approach uses different gain vectors for each phase while accounting for the jump. Although the probabilistic modeling of the jump may seem somewhat artificial in the case of a single-shot wing deployment operation, where no folding back of the wings is possible, the approach suggested can be thought of as a gain scheduling approach for systems with discrete operating points. This approach loses its advantage when the transition between these operating points is slow. The suggested procedure of comparing performance levels using both the deterministic and stochastic frameworks may be a useful addition to the overall control design process.

Although the application presented for the Markov jump approach, wing deployment in a UAV, is not very common in the aerospace industry, the design, analysis and simulation procedure suggested here may be relevant for other applications, where possibly fast enough transitions occur between discrete operating points.

Acknowledgment

The authors are indebted to Aron W. Pila for reading the manuscript, making numerous remarks, and improving it both in form and content.

References

[1] Gahinet, P., "Explicit Controller Formulas for LMI-Based H_∞ Synthesis," *Automatica*, Vol. 32, No. 7, 1996, pp. 1007–1014.

[2] Dragan, V., and Morozan, T., "Stability and Robust Stabilization to Linear Stochastic Systems Described by Differential Equations with Markovian Jumping and Multiplicative White Noise," Preprint Series, Inst. of Mathematics of the Romanian Academy, 1999, No. 17/1999.

[3] deFarias, D. P., Geromel, J. C., doVal, J. B. R., and Costa, O. L. V., "Output Feedback Control of Jump Markov Linear Systems in Continuous Time," *Transactions on Automatic Control*, Vol. AC-45, 2000, pp. 944–948.

[4] McRuer, D., Ashkenas, I., and Graham, D., *Aircraft Dynamics and Automatic Control*, Princeton Univ. Press, Princeton, NJ, 1973.

[5] Boyd, S., El Ghaoui, L., Feron, E., and Balakrishnan, V., *Linear Matrix Inequality in Systems and Control Theory*, Frontier Series, Society for Industrial and Applied Mathematics, Philadelphia, 1994.

[6] Gahinet, P., Nemirovsky, A., Laub, A. J., and Chilali, M., "LMI Control Toolbox for Use with MATLAB," Mathworks, Inc., Natick, MA, 1995.

Fast Quaternion Attitude Estimation from Two Vector Measurements

F. Landis Markley*
NASA Goddard Space Flight Center,
Greenbelt, Maryland 20771

Introduction

MANY spacecraft attitude-determination methods use exactly two vector measurements, e.g., unit vectors along the line of sight to a star or the sun or along the Earth's magnetic field. We want to find the attitude matrix that transforms vectors from some reference frame to the spacecraft body frame. That is, we would like to find a 3 × 3 proper orthogonal matrix A such that

$$A\boldsymbol{r}_i = \boldsymbol{b}_i \qquad \text{for} \qquad i = 1, 2 \tag{1}$$

where $\boldsymbol{b}_1$ and $\boldsymbol{b}_2$ are the measured unit vectors in the spacecraft body frame and $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$ are the corresponding unit vectors in the reference frame. It is impossible to satisfy both of these equations in general because they imply that $\boldsymbol{b}_1 \cdot \boldsymbol{b}_2 = \boldsymbol{r}_1 \cdot \boldsymbol{r}_2$, which might not be true in the presence of measurement errors. All reasonable two-vector attitude-determination schemes give the same estimate when this equality holds, however.

The earliest algorithm for determining spacecraft attitude from two vector measurements was the TRIAD algorithm,[1,2] which is simple to implement but does not treat the observations optimally. Wahba[3] proposed that the optimal attitude matrix should minimize the loss function[4]

$$L(A) \equiv \frac{1}{2}\sum_i a_i |\boldsymbol{b}_i - A\boldsymbol{r}_i|^2 = \sum_i a_i - \operatorname{trace}(AB^T) \tag{2}$$

where the a_i are positive weights assigned to the measurements and

$$B \equiv \sum_i a_i \boldsymbol{b}_i \boldsymbol{r}_i^T \tag{3}$$

Shuster showed a simplification of his optimal QUEST algorithm for the two-observation Wahba problem,[5] but Ref. 6 presented the first explicit closed-form optimal solution. Most existing optimal two-observation algorithms are significantly slower than TRIAD and actually slower than optimal n-observation algorithms. Recent exceptions are Mortari's optimal EULER-2 algorithm[7] and a suboptimal algorithm proposed by Reynolds.[8] The present Note presents two new algorithms for quaternion estimation from two vector measurements. The first is a very efficient optimal algorithm, which is almost as fast as the TRIAD algorithm. The second produces the same suboptimal estimate as TRIAD, but at reduced computational cost.

Received 11 June 2001; revision received 3 October 2001; accepted for publication 5 October 2001.

*Aerospace Engineer, Guidance, Navigation, and Control Systems Engineering Branch, Code 571. Fellow AIAA.