Table 1 J and n_a calculations by different reduction methods

Modes	Minreal		Schur		Ohkapp		Bst-Rem		Error
N	n_a	J_{MS}	n_a	J_{MS}	n_a	J_{MS}	n_a	J_{MS}	J
5	17	3.69	19	3.56	18	3.99	6	11.16	0.26
10	81	6.16	80	6.16	79	7.36	31	11.28	0.51
15	192	10.40	187	10.46	82	16.46	37	23.27	0.73
20	344	23.45	337	23.78	152	24.16	69	38.19	1.07

we construct an optimal approximation of $\boldsymbol{Z}(s)$. If we denote by $\hat{\boldsymbol{Q}}(s) = \boldsymbol{A}_0 + \boldsymbol{A}_1 s + \boldsymbol{A}_2 s^2 + \hat{\boldsymbol{Z}}(s)s$, then we have

$$\|\hat{\boldsymbol{Q}}(s) - \boldsymbol{Q}^{\mathrm{TAB}}\|_\infty \le \|\hat{\boldsymbol{Q}}(s) - \bar{\boldsymbol{Q}}(s)\|_\infty + \|\bar{\boldsymbol{Q}}(s) - \boldsymbol{Q}^{\mathrm{TAB}}\|_\infty$$
$$\le \|\bar{\boldsymbol{Q}}(s) - \boldsymbol{Q}^{\mathrm{TAB}}\|_\infty + \varepsilon$$

This shows that the additive model-order reduction $\boldsymbol{Z}(s)$ by $\hat{\boldsymbol{Z}}(s)$ only degrades the norm of the approximation by an order of ε. A similar inequality can be showed for the relative-multiplicative model-order reduction.

Numerical Results

A flexible aircraft with 5, 10, 15, and 20 vibration modes has been considered. The finite element model of the symmetric one-half of an aircraft is used to verify this new optimization theory of unsteady aerodynamic forces.[7] The DLM implemented in STARS[7] was used to obtain the tabulated unsteady aerodynamic matrices in the frequency domain, for a given Mach number $M = 0.8$ and a set of 14 reduced frequencies $k \in \{0.01, 0.1, 0.2, 0.303, 0.4, 0.5, 0.5882, 0.6250, 0.6667, 0.7143, 0.7692, 0.8333, 0.9091, 1.0000\}$. With these tabulated data each element of the aerodynamic matrix is approximated by a Padé approximant. The global error of the approximation of our method is reported in the last column of Table 1. Following the procedure developed in the preceding section, we construct a strictly proper rational matrix from the matrix of Padé approximants [Eq. (7)]. Considering the strictly proper rational matrix as the transfer function of a linear system, we construct a reduced-order model using different reduction methods from control theory.

Eliminating the unobservable and the uncontrollable states with the help of the Minreal function of MATLAB, we obtain a minimal realization of order n_a, as shown in the second column in Table 1. Following the results of the preceding section, we construct the two additive reduced-order models, using the Schur and the Optimal Hankel approximation method. The reduced-order model for the latter two methods are reported in columns 4 and 6 of Table 1, respectively. The eighth column represents the dimension of the reduced system obtained using the balanced stochastic truncation method. We can see that this last method gives the best results in terms of aerodynamic dimension n_a in comparison to the other methods. The tolerance used in the calculation of the minimal realization and the approximation of the reduced-order models was chosen to be 10^{-6}. The aerodynamic dimensions n_a found by the four methods are now used to perform the MS approximation of the unsteady aerodynamic forces. This is done in order to compare the errors of unsteady aerodynamic forces approximation for a fixed aerodynamic dimension n_a. The errors for the MS method, denoted by J_{MS} for each method, are reported in columns 3, 5, 7, and 9 of Table 1. It can be seen that the approximation error calculated by our method is 12–40 times smaller than the errors calculated by the MS procedure.

Conclusions

The main contribution of this Note is an original method for the approximation of the unsteady aerodynamic forces using recent results from linear system theory and Padé approximants. Starting with a Padé approximation of the unsteady aerodynamic forces, we construct a reduced-order model for stability analysis purposes. Contrary to the standard MS approximation, the error of the approximation is independent on the aerodynamic dimension of the final aeroelastic system. Running the MS procedure with different numbers of lags to get a good approximation can take large amounts of time because the procedure is highly iterative. Our method overcomes the problem of choosing the number of lags (aerodynamic dimension) n_a of the MS procedure because in our method the aerodynamic dimension is a result not an initial parameter. Furthermore, our method yields better approximation error.

Acknowledgments

We would like to thank Vivek Mukhopadhyay for his comments and remarks on this Note. The authors would also like to thank Kajal Gupta at NASA Dryden Flight Research Center for allowing us to use STARS program and the Aircraft Test Model ATM. Many thanks are owed to the other members of the STARS Engineering group for their continuous assistance and collaboration: Tim Doyle, Can Bach, and Shun Lung.

References

[1]Tiffany, S. H., and Adams, W. M., "Nonlinear Programming Extensions to Rational Function Approximation of Unsteady Aerodynamics," NASA TP-2776, July 1988.

[2]Edwards, J. W., "Unsteady Aerodynamic Modeling and Active Aeroelastic Control," Stanford Univ., SUDAAR 504, Stanford, CA, Feb. 1977.

[3]Roger, K. L., "Airplane Math Modeling Methods for Active Control Design," *Structural Aspects of Active Controls*, CP-228, AGARD, Aug. 1977, pp. 4-1–4-11.

[4]Vepa, R., "Finite State Modeling of Aeroelastic System," NASA CR-2779, Feb. 1977.

[5]Karpel, M., "Design for Flutter Suppression and Gust Alleviation Using State Space Modeling," *Journal of Aircraft*, Vol. 19, No. 3, 1982, pp. 221–227.

[6]Baker, G. A., *Padé Approximants*, Cambridge Univ. Press, Cambridge, England, U.K., 1996, pp. 1–746.

[7]Gupta, K. K., "STARS-An Integrated Multidisciplinary, Finite Element, Structural, Fluids, Aeroelastic and Aeroservoelastic Analysis Computer Program," NASA TM-4795, 1997, pp. 1–285.

Composite Optimization Scheme for Time-Optimal Control

Michael C. Reynolds* and Peter H. Meckl†
Purdue University, West Lafayette, Indiana 47907-1288

I. Introduction

ALTHOUGH there are many techniques used to solve optimization problems, most of them can be categorized as constrained optimization methods. Constrained optimization involves the development and minimization of a cost function subject to a set of weighted constraints. These techniques are popular because they are usually easy to set up and they can be solved with many standard computational packages. But without a quality initial guess for the optimization routine, there might be no convergence or convergence to an undesirable solution. Finding a quality initial guess can be a difficult task in large systems of equations. A more systematic approach to finding an initial guess can be found by using linear programming. Linear programming can find the optimal solution to a system of equations without an initial guess. Starting with either linear or nonlinear equations, the problem is converted into a linear discrete form whose solution approximates the solution of the original problem. As the discrete intervals become smaller, the result approaches the continuous solution. The largest drawback of the linear programming technique is the lengthy computation time.

Received 31 July 2001; revision received 14 March 2002; accepted for publication 13 April 2002.

*Doctoral Candidate, School of Mechanical Engineering; reynoldm@purdue.edu.

†Associate Professor, School of Mechanical Engineering; meckl@ecn.purdue.edu.

In this Note we show how combining linear programming and constrained optimization into a composite process can solve a common motion control problem more efficiently.

II. Development of Time-Optimal Control Problem

A common system model for point-to-point motion control algorithms is the two-mass floating oscillator.[1,2] The objective is to find the time-optimal input such that the unforced endpoint mass moves a distance x_f. Typically both the initial and final states are constrained to be rest states, which is the case in this analysis. It is also assumed that the input is bounded by a maximum force $F_{\max}$. The motion control problem is then characterized by the following dimensionless variable:

$$\omega_n T_s = \sqrt{\frac{4(m_1 + m_2)x_f\omega_n^2}{F_{\max}}} \tag{1}$$

where mass m_1 is the forced mass, m_2 is the unforced mass, and ω_n is the undamped natural frequency. Dimensionless time is defined by $t^* = \omega_n t$, where t is dimensional time. The $\omega_n T_s$ value is simply the dimensionless rigid-body move time. The quantity $\omega_n T_s/2\pi$ represents the number of natural periods that occur during the duration of the command profile. This dimensionless quantity gives a lower bound on move time for all possible inputs because no input having the same peak force constraint $F_{\max}$ can move the system a distance x_f in less time. The $\omega_n T_s$ value is also important in evaluating the time efficiency of a particular control strategy as it can be used as a benchmark to compare control designs.[1] If a certain input has a move time that is very close to $\omega_n T_s$, one could assume that further control optimization would produce diminishing results. Systems whose move times are much greater than $\omega_n T_s$ are often excellent candidates for command shaping techniques.[1,3]

III. Constrained Optimization Formulation

To solve the time-optimal control problem within a constrained optimization package, the constraint equations need to be defined. The goal is to minimize the final time t_f subject to the following constraints:

$$1 - 2\cdot\sum_{i=1}^{n}(-1)^i \exp(\zeta\omega_n t_i)\cos(\omega_d t_i) + \exp(\zeta\omega_n t_f)\cos(\omega_d t_f) = 0 \tag{2}$$

$$2\cdot\sum_{i=1}^{n}(-1)^{(i+1)} \exp(\zeta\omega_n t_i)\sin(\omega_d t_i) - \exp(\zeta\omega_n t_f)\sin(\omega_d t_f) = 0 \tag{3}$$

$$\frac{1}{2}\left[(\omega_n t_f)^2 + 2\cdot\sum_{i=1}^{n}(-1)^i(\omega_n t_f - \omega_n t_i)^2\right] = \left(\frac{\omega_n T_s}{2}\right)^2 \tag{4}$$

$$\sum_{i=1}^{n}(-1)^{(i+1)}t_i - \frac{t_f}{2} = 0 \tag{5}$$

where t_i are the n switch times, t_f is the final switch, ζ is the damping ratio, and $\omega_d = \omega_n\sqrt{(1-\zeta^2)}$. It is assumed that the initial switch occurs at zero time and is in the positive direction. Equations (2) and (3) represent the cancellation of the flexible mode, whereas Eqs. (4) and (5) ensure that the rigid-body mode undergoes the desired point-to-point motion. Although these four equations fit nicely into most computational packages, the most critical element is the initial guess for the solution of switch times t_i and t_f.

The initial guess depends on the number of switches (both internal and final) and their location. Tuttle and Seering[4] as well as Liu and Wie[5] have suggested that a good guess for the number of internal switches is the number of system poles minus one. It is also important to know how to estimate the location of the internal switches because guessing the proper number of switches does not guarantee convergence to the optimal solution. In the undamped case these switches are symmetric about the midmove time,[6] but with damping it can be difficult to estimate their locations. Therefore, a more systematic approach is needed.

IV. Linear Programming Formulation

Pao[6] has shown an alternative formulation to the time-optimal control problem that uses constrained linear programming. Using a discrete-time model, the objective is to find an input, bounded by the actuator limits, that reduces the norm of the state vector to zero at the end of the control time. The discrete-time model is given by

$$\boldsymbol{x}_{k+1} = \Phi\boldsymbol{x}_k + \Gamma u_k \tag{6}$$

$$\Phi = e^{AT^*} \tag{7}$$

$$\Gamma = \int_0^{T^*} e^{A\xi}\,\mathrm{d}\xi\cdot\boldsymbol{b} \tag{8}$$

where $\boldsymbol{x}_k$ is the dimensionless state vector of the system at time index k, T^* is the dimensionless sampling period ($T^* = \omega_n T$, where T is the sampling period in seconds), and $\boldsymbol{A}$ and $\boldsymbol{b}$ are from the dimensionless state-space model $\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}u$. The product kT^* represents dimensionless move time.

The problem fits well into typical linear programming packages. The biggest advantage of linear programming is that the number of switches does not need to be built into the program. The discrete input vector directly sketches out the input force signal, and, therefore, the number of switches between peak positive and peak negative force and their approximate locations can be easily determined. The precision of the switches is tightly related to the sampling time of the discrete system. The biggest drawback of the linear programming algorithm is its computational inefficiency. Although the computational time for constrained optimization is on the order of a few seconds, getting precise results from linear programming can take many hours.

V. Composite Solution Approach

The necessity of repeatedly using linear programming to find the time-optimal command profile is a large drawback to the linear programming solution method. Another drawback is the intense CPU effort required for finding precise switch times. Combining both the constrained optimization and linear programming methods into one solution process can eliminate these problems. By using a large sampling time, a crude approximation can be made of the switching profile using linear programming. This approximation can be used to generate initial guesses for the switch times in the constrained optimization solver. In our work we found that starting with a T^* of 0.1 in the linear program normally ensures enough precision in the switch guesses to converge in the constrained optimization routine. Because T^* is equal to the product of the natural frequency and dimensional sample period, having a maximum T^* of 0.1 ensures at least 50 sample points per period of oscillation. (If this generates switch times that do not lead to constrained optimization convergence, T^* is reduced, and the linear program is rerun.) The results of the linear program are then fed into a standard constrained optimization routine as an initial guess. The final step of the composite process is to verify the optimality of the solution.[7,8]

VI. Testing the Efficiency of the Composite Approach

Testing was performed in order to show the improved efficiency of the composite approach. We compared the composite technique to a constrained optimization alone with random guesses. The random guesses had to be bounded by some maximum time, which was set to be four times the rigid-body move time because solutions that cancel flexible modes can take as long as four times the rigid-body move time for systems with $\omega_n T_s$ greater than one. These random guesses were then fed into a standard MATLAB® constrained optimization routine. If the result was nonconvergence to the optimal solution, a new random guess was made. The program would continue this process until the result converged and the solution was verified to be time optimal. With random guessing each run consisted of numerous initial guesses that eventually would give one that worked. The results given in Table 1 are an average of 20 runs

Table 1 Run times in seconds for Random guessing and composite technique using 1-GHz processor with MATLAB 6.0 on Windows 2000 platform

ζ	$\omega_n T_s = 2$	$\omega_n T_s = 15$	$\omega_n T_s = 30$
		Random guessing	
0	0.38	511.0	1104
0.15	1.00	18.64	326.5
0.5	0.39	12.33	2338
		Composite technique	
0	12	55	240
0.15	10	52	230
0.5	10	45	360

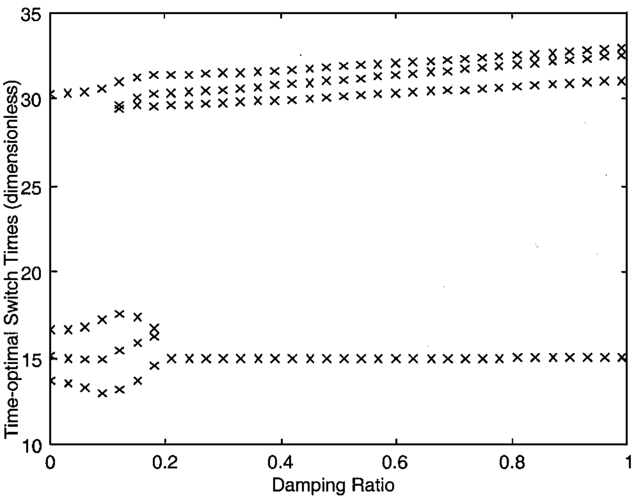

Fig. 1 Dimensionless time-optimal switch times for $\omega_n T_s = 30$ as a function of damping ratio. Each × represents an optimal switch time. Notice how the switches are often bunched together.

of both the random guessing and the composite process. These results were all performed on a 1-GHz processor running MATLAB 6.0. (We developed a MATLAB GUI for the composite process, which is available via E-mail to the authors.) Although the absolute times might vary, it is the relative performance of the two techniques that should be emphasized. The results show that for low $\omega_n T_s$ values the random guessing procedure was actually more efficient, but for higher values of $\omega_n T_s$ the composite technique was faster. To put this in perspective, a typical motion command in a high-performance manufacturing process had $\omega_n T_s = 35.5$ (Ref. 9). The cases that were the worst for random guessing were ones in which the optimal switch times were packed tightly near each other (Fig. 1). This makes sense as random guesses will tend to create more symmetric switch times. Solving for the complete set of results shown in Table 1 using random guesses took four times longer than the composite approach. The run times of the composite technique were generally not a function of damping, but for the case of $\omega_n T_s = 30$ and $\zeta = 0.5$ the T^* value had to be decreased, explaining why it took slightly longer. The run times of the composite approach were nearly all caused by the linear programming phase. If linear programming alone had been used, a smaller T^* would have been necessary for the same precision on the final switch times, therefore requiring more time to complete than the composite approach.

VII. Conclusions

Using linear programming to generate initial guesses for constrained optimization was more efficient than using a constrained optimization approach with random guesses for the time-optimal slewing of the floating oscillator. Thus, in the absence of good initial guesses it would be more efficient to use the composite approach. Although the results are restricted to this particular problem, we would suggest this method for other problems where initial guesses that lead to optimal convergence are difficult to generate.

References

[1]Reynolds, M. C., and Meckl, P. H., "Benchmarking Time-Optimal Control Inputs for Flexible Systems," *Journal of Guidance, Control, and Dynamics*, Vol. 25, No. 2, 2002, pp. 215–221.

[2]Scrivener, S. L., and Thompson, R. C., "Survey of Time-Optimal Attitude Maneuvers," *Journal of Guidance, Control, and Dynamics*, Vol. 17, No. 2, 1994, pp. 225–233.

[3]Singer, N., and Seering, W., "Preshaping Command Inputs to Reduce System Vibrations," *Journal of Dynamic Systems, Measurement and Control*, Vol. 112, No. 1, 1990, pp. 76–82.

[4]Tuttle, T. D., and Seering, W. P., "Creating Time-Optimal Commands for Systems with Denominator Dynamics," *Proceedings of the Institute of Electrical and Electronics Engineers Conference on Control Applications*, Inst. of Electrical and Electronics Engineers, Piscataway, NJ, 1996, pp. 385–390.

[5]Liu, Q., and Wie, B., "Robust, Time-Optimal Control of Uncertain Flexible Spacecraft," *Journal of Guidance, Control, and Dynamics*, Vol. 15, No. 3, 1992, pp. 597–604.

[6]Pao, L. Y., "Minimum-Time Control Characteristics of Flexible Structures," *Journal of Guidance, Control, and Dynamics*, Vol. 19, No. 1, 1996, pp. 123–129.

[7]Pao, L. Y., and Singhose, W. E., "Verifying Robust Time-Optimal Commands for Multimode Flexible Spacecraft," *Journal of Guidance, Control, and Dynamics*, Vol. 20, No. 4, 1997, pp. 831–833.

[8]Ben-Asher, J., Burns, J. A., and Cliff, E. M., "Time Optimal Slewing of Flexible Spacecrafts," *Journal of Guidance, Control, and Dynamics*, Vol. 15, No. 2, 1992, pp. 360–367.

[9]Meckl, P. H., and Umemoto, K., "Achieving Fast Motions in Semiconductor Manufacturing Machinery," *Proceedings of Mechatronics 2000*, Elsevier Science, Oxford, 2000.

Height Control System for Sea-Skimming Missile Using Predictive Filter

S. E. Talole* and S. B. Phadke*
Institute of Armament Technology, Pune 411 025, India

Introduction

A SEA-SKIMMING missile flies very low above mean sea level and normally uses a radio altimeter for measuring its altitude from sea surface. Because the altimeter measures the actual height from sea surface rather than the height from mean sea level, the sea waves effectively act as a disturbance making the missile follow a trajectory parallel to the sea surface resulting in unnecessary expense of missile energy and increasing the chances of missile ditching into the sea. The objective of this Note is to present a height control system employing a filter based on a recently developed nonlinear predictive filter algorithm to estimate the height of the missile above the mean sea level. Stability analysis of the closed-loop system using the predictive filter is carried out. The simulation results show that the performance of the height control system using the filter adequately meets the objectives of a sea-skimming missile.

Height Control System

Sea-skimming missiles are an effective weapon against ships as they fly very low (typically less than 10 m) above the mean sea level remaining below the horizon of the enemy radar on the one hand and causing substantial damage to the enemy ships on the other. These missiles normally use an altimeter to sense their altitude above the

Received 7 January 2002; revision received 27 April 2002; accepted for publication 6 May 2002.

*Scientist, GM Faculty, Institute of Armament Technology, Girinagar.