JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 28, No. 3, May–June 2005

Engineering Notes

ENGINEERING NOTES are short manuscripts describing new developments or important results of a preliminary nature. These Notes should not exceed 2500 words (where a figure or table counts as 200 words). Following informal review by the Editors, they may be published within a few months of the date of receipt. Style requirements are the same as for regular contributions (see inside back cover).

Nonlinear Control Allocation Using Piecewise Linear Functions: A Linear Programming Approach

Michael A. Bolender* and David B. Doman†
U.S. Air Force Research Laboratory, Wright–Patterson Air Force Base, Ohio 45433

Introduction

HISTORICALLY, the control allocation problem has been approached by assuming that aerodynamic control effectors produce control moments that are linear functions (at least locally) of the control surface deflection. Whereas this approach may be acceptable for nearly all aircraft under nominal flight conditions when there are no failed control surfaces, the occurrence of one or more failed control effectors may result in a situation where the linear assumption is no longer valid. This is because the dominant effects of most control surfaces are approximately linear over normal operating ranges in at least one axis; however, off-axis effects can be nonlinear. When a control effector fails, the resulting rolling, pitching, and yawing moments produced by the failed effector must be accommodated by reconfiguring the remaining healthy control effectors. For some failures, one or more of the remaining effectors may be driven toward a position limit to compensate for the failed effector. The behavior of the control moment curve for many aircraft in regions near position limits tends become highly nonlinear. Furthermore, the failure of a major linear effector in one axis may lead to the necessity of using secondary nonlinear effects that are normally treated as disturbances by the control system. Therefore, the linear model may introduce significant modeling errors that result in incorrect control surface deflections, and this may ultimately lead to loss of the vehicle.

Recently, Bolender and Doman[1] have shown that a piecewise linear representation of the control moment curve accounts for the nonlinearities inherent in aerodynamic data. The only requirement is that the control moments generated by each effector be separable, meaning that no aerodynamic interference occurs among the effectors. The resulting control allocation problem was posed as a mixed-integer linear program and solved using a public domain branch-and-bound optimization code. The downside of this approach was the length of time needed by the solver to find a solution, making implementation in a digital flight-control system impracticable.

The intent of this Note is to show that the piecewise linear control allocation problem can be solved fast enough to be implemented in a digital flight-control system. The approach taken here differs from the authors' initial paper on this subject in two ways. The first difference is a move away from the mixed-integer linear programming form of the optimization problem and to a linear programming formulation. The linear programming problem will be solved using a modified simplex algorithm where a rule-based approach is employed to enforce the necessary adjacency constraints on the interpolating coefficients. The second difference is that we solve a mixed optimization problem[2] as opposed to the solution of the multibranch control allocation problem.[3] This allows us to achieve the same objective as before, but only having to solve one optimization problem instead of two. We will compare the performance of the simplex method with restricted basis entry rules to the mixed-integer formulation and show that the two approaches give equivalent solutions to the same set of control allocation problems. To perform this comparison, we will look at both closed-loop and open-loop control allocation problems. In the latter case, a set of control allocation problems are randomly selected and solved by each approach. For the former, we compare the algorithms in a digital simulation of a reentry vehicle on approach and landing.

Piecewise Linear Mixed Optimization Control Allocation

The control allocation problem solved in Ref. 1 used two distinct performance indices that were similar to those used in Buffington's[3] multibranch approach. However, instead of a linear relationship between the control effector displacements and the control moments, a nonlinear relationship was used. Let the nonlinear vector-valued function $\boldsymbol{G}(\boldsymbol{\delta})$ denote the relationship between the control effector positions and their moments. The function $\boldsymbol{G}(\boldsymbol{\delta})$ maps $\mathbb{R}^n$ into $\mathbb{R}^m$ where $n \geq m$, where n is the number of control effectors and m is the number of controlled variables. Let $\boldsymbol{d}_{\text{des}}$ denote the controlled variables. In this case, the controlled variables are the rolling, pitching, and yawing moments. Buffington's multibranch approach requires that two optimization problems be solved. The first optimization problem is called the control deficiency branch. The objective of the control deficiency branch is to minimize $\|\boldsymbol{W}_a(\boldsymbol{G}(\boldsymbol{\delta}) - \boldsymbol{d}_{\text{des}})\|_1$ subject to position and rate constraints on $\boldsymbol{\delta}$. The value of the performance index for this optimization problem indicates whether or not $\boldsymbol{d}_{\text{des}}$ is feasible. If feasibility of the control allocation problem has been ascertained, a second optimization problem is then solved that minimizes some secondary objective. The objective function of this second optimization is typically taken to be $\|\boldsymbol{W}_u(\boldsymbol{\delta} - \boldsymbol{\delta}_p)\|_1$ subject to $\boldsymbol{G}(\boldsymbol{\delta}) = \boldsymbol{d}_{\text{des}}$ and position and rate constraints on $\boldsymbol{\delta}$. This is commonly referred to as the control sufficiency branch.

The mixed optimization problem that was formulated by Bodson[2] combines the two branches of the multibranch control allocation problem into a single optimization problem. A new parameter ϵ is introduced for the purpose of prioritizing either control deficiency or control sufficiency. The mixed optimization problem is stated as

$$\min J = \|\boldsymbol{W}_a(\boldsymbol{G}(\boldsymbol{\delta}) - \boldsymbol{d}_{\text{des}})\|_1 + \epsilon\|\boldsymbol{W}_u(\boldsymbol{\delta} - \boldsymbol{\delta}_p)\|_1 \tag{1}$$

subject to

$$\boldsymbol{\delta}_{\min} \leq \boldsymbol{\delta} \leq \boldsymbol{\delta}_{\max} \tag{2}$$

where $\boldsymbol{W}_a = \text{diag}(w_{a1}, w_{a2}, w_{a3}, \ldots, w_{am})$ is a weighting matrix used to prioritize a given control axis; $\boldsymbol{G}(\boldsymbol{\delta})$ is a nonlinear, vector-valued function that maps $\mathbb{R}^n$ to $\mathbb{R}^m$; $\boldsymbol{\delta}$ is an $n \times 1$ vector of control

Presented as Paper 2004-5019 at the AIAA Guidance, Navigation, and Control Conference, Providence, RI, 16–19 August 2004; received 23 August 2004; revision received 16 November 2004; accepted for publication 22 November 2004.

*Aerospace Engineer, Control Design and Analysis Branch, 2210 Eighth Street, Room 305. Senior Member AIAA.

†Senior Aerospace Engineer, Control Design and Analysis Branch, 2210 Eighth Street, Room 305. Associate Fellow AIAA.

effectors; $\boldsymbol{W}_u = \text{diag}(w_{u1}, w_{u2}, w_{u3}, \ldots, w_{un})$ is a weighting matrix on the control effectors; and $\boldsymbol{\delta}_p$ is an $n \times 1$ vector of preferred control effector displacements. Again, it is assumed that $n \geq m$. For the moment we will make the assumption that $\boldsymbol{G}(\boldsymbol{\delta}) = \boldsymbol{B}\boldsymbol{\delta}$. The mixed optimization problem can then be posed as a linear programming problem. The corresponding linear program can then be solved by any readily available linear programming software.

Bodson[2] gives one possible transformation to a linear programming problem for the optimization problem defined in Eqs. (1) and (2); however, we selected a transformation approach that can be found in Ref. 4. The transformation relies on the observation that $|x|$ is the smallest number x_s that satisfies both $x \leq x_s$ and $-x \leq x_s$. As a result, we are able to pose the mixed optimization problem as follows:

$$\min J = \boldsymbol{W}_a\boldsymbol{\delta}_s + \epsilon\boldsymbol{W}_u\boldsymbol{u}_s \tag{3}$$

subject to

$$\boldsymbol{\delta}_s \geq 0 \tag{4}$$

$$\boldsymbol{u}_s \geq 0 \tag{5}$$

$$\boldsymbol{B}\boldsymbol{\delta} + \boldsymbol{\delta}_s \geq \boldsymbol{d}_{\text{des}} \tag{6}$$

$$-\boldsymbol{B}\boldsymbol{\delta} + \boldsymbol{\delta}_s \geq -\boldsymbol{d}_{\text{des}} \tag{7}$$

$$\boldsymbol{\delta} + \boldsymbol{u}_s \geq \boldsymbol{\delta}_p \tag{8}$$

$$-\boldsymbol{\delta} + \boldsymbol{u}_s \geq -\boldsymbol{\delta}_p \tag{9}$$

$$\boldsymbol{\delta}_{\min} \leq \boldsymbol{\delta} \leq \boldsymbol{\delta}_{\max} \tag{10}$$

The vectors $\boldsymbol{\delta}_s$ and $\boldsymbol{u}_s$ are vectors of slack variables. The reason for selecting this particular transformation as opposed to the one by Bodson[2] is that this formulation allows us to easily implement the piecewise linear function approximation.

To convert the linear programming problem into a piecewise linear programming problem, we simply replace $\boldsymbol{B}\boldsymbol{\delta}$ with a piecewise linear representation of each control moment curve, that is, $L_i(\delta_i)$, $M_i(\delta_i)$, and $N_i(\delta_i)$. We choose a set of breakpoints for each δ_i, $i = 1, \ldots, n$, such that

$$\delta_i = \sum_{k=1}^{K_i} \lambda_i^{(k)} \delta_i^{(k)} \tag{11}$$

$$\sum_{k=1}^{K_i} \lambda_i^{(k)} = 1 \tag{12}$$

$$\lambda_i^{(j)}\lambda_i^{(k)} = 0, \quad \text{if} \quad k > j+1, \quad j = 1, \ldots, K_i - 1 \tag{13}$$

where $\lambda_i^{(k)}$ is a nonnegative interpolating coefficient corresponding to the ith control effector at breakpoint k, and K_i denotes the number of breakpoints for the ith control effector. Equation (13) is necessary to ensure that δ_i is approximated by no more than two adjacent values of $\lambda_i^{(k)}$. If δ_i falls at a breakpoint, there will only be one value of $\lambda_i^{(k)}$ that is nonzero and Eq. (13) is still valid.

The piecewise linear approximations for the control moments as a function of δ_i are written as

$$L_i = L(\delta_i) = \sum_{k=1}^{K_i} \lambda_i^{(k)} L_i^{(k)} \tag{14}$$

$$M_i = M(\delta_i) = \sum_{k=1}^{K_i} \lambda_i^{(k)} M_i^{(k)} \tag{15}$$

$$N_i = N(\delta_i) = \sum_{k=1}^{K_i} \lambda_i^{(k)} N_i^{(k)} \tag{16}$$

where $L_i^{(k)}$, $M_i^{(k)}$, and $N_i^{(k)}$ are the values of the rolling, pitching, and yawing moment curves evaluated at the kth breakpoint for the ith control effector. We are now able to replace $\boldsymbol{B}\boldsymbol{\delta}$ with $\tilde{\boldsymbol{B}}\boldsymbol{\Lambda}$, where

$$\tilde{\boldsymbol{B}} = \begin{bmatrix} L_1^{(1)} & L_1^{(2)} & \ldots & L_i^{(k)} & \ldots & L_n^{(K_n)} \\ M_1^{(1)} & M_1^{(2)} & \ldots & M_i^{(k)} & \ldots & M_n^{(K_n)} \\ N_1^{(1)} & N_1^{(2)} & \ldots & N_i^{(k)} & \ldots & N_n^{(K_n)} \end{bmatrix} \tag{17}$$

$$\boldsymbol{\Lambda} = \begin{bmatrix} \lambda_1^{(1)} \\ \lambda_1^{(2)} \\ \vdots \\ \lambda_i^{(k)} \\ \vdots \\ \lambda_n^{(K_n)} \end{bmatrix} \tag{18}$$

The vector $\boldsymbol{\Lambda}$ is of length

$$\sum_{i=1}^{n} K_i$$

and $\tilde{\boldsymbol{B}}$ is a matrix of size

$$n_c \times \sum_{i=1}^{n} K_i$$

where n_c is the number of controlled variables. In the piecewise linear optimization problem, the constraints $\boldsymbol{\delta}_{\min} \leq \boldsymbol{\delta} \leq \boldsymbol{\delta}_{\max}$ are replaced by $\lambda_i^{(k)} \geq 0$. The upper and lower bounds on $\boldsymbol{\delta}$ are accounted for in the selection of the breakpoints for each δ_i. Once we obtain an optimal solution to the problem, we compute each δ_i using Eq. (11). It is also necessary to include in the problem the n constraints that correspond to Eq. (12).

The resulting optimization problem is

$$\min J = \boldsymbol{W}_a\boldsymbol{\delta}_s + \epsilon\boldsymbol{W}_u\boldsymbol{u}_s \tag{19}$$

subject to

$$\boldsymbol{\delta}_s \geq 0 \tag{20}$$

$$\boldsymbol{u}_s \geq 0 \tag{21}$$

$$\tilde{\boldsymbol{B}}\boldsymbol{\Lambda} + \boldsymbol{\delta}_s \geq \boldsymbol{d}_{\text{des}} \tag{22}$$

$$-\tilde{\boldsymbol{B}}\boldsymbol{\Lambda} + \boldsymbol{\delta}_s \geq -\boldsymbol{d}_{\text{des}} \tag{23}$$

$$\sum_{k=1}^{K_i} \lambda_i^{(k)} \delta_i^{(k)} + u_{s,i} \geq \delta_{p,i}, \qquad i = 1, \ldots, n \tag{24}$$

$$-\sum_{k=1}^{K_i} \lambda_i^{(k)} \delta_i^{(k)} + u_{s,i} \geq -\delta_{p,i}, \qquad i = 1, \ldots, n \tag{25}$$

$$\lambda_i^{(k)} \geq 0, \qquad i = 1, \ldots, n, \qquad k = 1, \ldots, K_i \tag{26}$$

$$\sum_{k=1}^{K_i} \lambda_i^{(k)} = 1, \qquad i = 1, \ldots, n \tag{27}$$

$$\lambda_i^{(j)}\lambda_i^{(k)} = 0, \qquad \text{if} \qquad k > j+1, \qquad j = 1, \ldots, K_i - 1 \tag{28}$$

The constraint given by Eq. (28) forces us to choose one of two approaches to obtain a solution to the preceding optimization problem. The first approach is to define a set of binary variables along

with a set of additional constraints that enforce Eq. (28). The resulting optimization problem is then a mixed-integer linear program. The second approach is to solve the optimization problem as a linear programming problem using a modified simplex algorithm. To accommodate Eq. (28), we employ a simple rule-based approach to determine which $\lambda_i^{(k)}$ are allowed to enter the basis.

Linear Program with Restricted Basis Entry Rules

We desire to solve the preceding optimization problem as a linear programming problem while imposing the constraint given by Eq. (28). In Ref. 1, Eq. (28) was enforced by introducing binary variables and appending a set of constraints to the problem formulation. Note, however, that if Eq. (28) were not present, we would have a linear programming problem that could be easily solved using the simplex method. Therefore, as an alternative to the mixed-integer linear programming formulation given in Ref. 1, we will enforce Eq. (28) by modifying the the simplex algorithm such that we only admit a $\lambda_i^{(k)}$ into the basis if Eq. (28) is satisfied, otherwise a new $\lambda_i^{(k)}$ is found. These rules are commonly referred to as restricted basis entry rules.[5] A description of the simplex algorithm with restricted basis entry rules is given hereafter in the algorithm. More details on the simplex algorithm with restricted basis entry rules can be found in Refs. 6 and 7.

To demonstrate the application of the restricted basis entry rules, we consider the simple example given here.

Example

Consider the following optimization problem:

$$\min x_1 + |x_2| \tag{29}$$

subject to

$$2x_1 + x_2 = 3 \tag{30}$$

$$x_1 \geq 0 \tag{31}$$

$$-1 \leq x_2 \leq 1 \tag{32}$$

To convert this problem to a piecewise linear program that is to be solved using the simplex algorithm with restricted basis entry rules, we choose the following breakpoints for $x_2 = \{-1, 0, 1\}$. Then $x_2 = -\lambda^{(1)} + 0\lambda^{(2)} + \lambda^{(3)}$ and $|x_2| = \lambda^{(1)} + 0\lambda^{(2)} + \lambda^{(3)}$. The transformed optimization problem is then

$$\min x_1 + \lambda^{(1)} + 0\lambda^{(2)} + \lambda^{(3)} \tag{33}$$

subject to

$$\tag{34}$$

$$2x_1 - \lambda^{(1)} + \lambda^{(3)} = 3 \tag{35}$$

$$\lambda^{(1)} + \lambda^{(2)} + \lambda^{(3)} = 1 \tag{36}$$

$$x_1 \geq 0 \tag{37}$$

$$\lambda^{(1)}, \lambda^{(2)}, \lambda^{(3)} \geq 0 \tag{38}$$

We define the decision vector $\boldsymbol{x} = [x_1 \ \ \lambda^{(1)} \ \ \lambda^{(2)} \ \ \lambda^{(3)}]^T$. Then $\Sigma = \Sigma_1 = \{2, 3, 4\}$ is the set of indices of those elements of the decision vector that must obey the restricted basis entry rules. Assume that we have x_1 and $\lambda^{(1)}$ in the basis; then $n_\Sigma = 1$. We compute the reduced cost $\bar{\boldsymbol{c}} = [0 \ \ 0 \ \ -1 \ \ 0]$. According to $\bar{\boldsymbol{c}}$, the only variable that is eligible to enter into the basis is $\lambda^{(2)}$ because it is the only element in the set of nonbasic variables that will reduce the cost function. Given that $\lambda^{(2)} \in \Sigma$, it must obey the restricted basis entry rules. Next, we determine that $\lambda^{(1)}$ will exit the basis. According to the algorithm, $\lambda^{(2)}$ will enter into the basis because the restricted basis entry rules are not violated. In this case, $\lambda^{(2)}$ enters and $\lambda^{(1)}$ exits, keeping a single element of Σ in the basis.

Note that the adjacency constraint does not appear explicitly in the optimization problem, but is enforced by careful implementation of the solution procedure.

A survey of commercial linear programming solvers indicated that simplex-based solvers with restricted basis entry rules were unavailable. Instead, the GNU Linear Programming Kit (GLPK), which is an open-source general linear program and mixed-integer linear program solver written in C, was modified to accommodate restricted basis entry rules.

Algorithm: Restricted Basis Entry Rules

Assume that we are given the following piecewise linear program: $\min \boldsymbol{c}^T\boldsymbol{x}$ subject to $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$. We partition $\boldsymbol{x}$ such that $\lambda_i^{(k)}$, $k = 1, \ldots, K_i$ are adjacent. Let Σ denote those elements of $\boldsymbol{x}$ that must obey the restricted basis entry rules. Furthermore, partition $\Sigma = \{\Sigma_1, \Sigma_2, \ldots, \Sigma_n\}$ such that Σ_i corresponds to the $\lambda_i^{(k)}$. Let Θ be the set of nonbasic variables that have negative reduced costs. Furthermore, assume that x_i is the next variable that has been chosen to enter the basis $\boldsymbol{\mathcal{B}}$, according to the selected pricing strategy. Let x_e denote the variable that will exit the basis if x_i were to enter.

```
While Θ ≠ ∅ do
  if x_i ∈ Σ then
    Let Σ_k denote the partition of Σ to which x_i belongs. Let
    n_Σ denote the cardinality of Σ_k ∩ B. Define x_j,
    j = 1, ..., n_Σ, as those elements of x that are in Σ_k ∩ B.
    if n_Σ = 0 then
      x_i enters the basis
      else if n_Σ = 1 then
        if j = i − 1 or j = i + 1 then
          x_i enters the basis and x_e exits the basis
        else if j ≠ i − 1 or j ≠ i + 1 and x_j = x_e then
          x_i enters the basis and x_e exits the basis
        else if j ≠ i − 1 or j ≠ i + 1 and x_j ≠ x_e then
          x_i does not enter the basis
        end if
    else if n_Σ = 2 then
        if j = {i − 2, i − 1} and x_e = x_{i−2} then
          x_i enters the basis and x_e exits the basis
        else if j = {i + 1, i + 2} and x_e = x_{i+2} then
          x_i enters the basis and x_e exits the basis
        else
          x_i does not enter the basis
        end if
    end if
  else if x_i ∉ Σ then
    x_i enters the basis and x_e exits the basis
  end if
  if x_i ∈ Σ and does not enter the basis then
    Choose the next x_i that enters the basis from the current set Θ
  else if x_i enters the basis then
    Update basis and compute new reduced costs
  end if
end while
```

Results

In this section, we show that the mixed-optimization problem implemented with piecewise linear approximations of the control moments and solved using a simplex method with restricted basis entry rules gives a solution that is comparable to that given by Bolender and Doman,[1] where a mixed-integer linear programming formulation was used. We also demonstrate that the mixed-optimization problem can be solved fast enough that it is a candidate for future use on a digital flight-control computer equipped with a processor that is comparable to one found on current desktop computers. For this study, the GLPK solver was implemented as a C mex file in MATLAB® and compiled to optimize execution speed. The computer on which this analysis was run was equipped with an Athlon 1800XP + processor, 1.5 GB of RAM, and the Windows 2000 operating system. We compare the execution time of the simplex method with restricted basis entry rules to the time required to solve the same

optimization problem using the mixed-integer linear programming formulation of the mixed-optimization problem. The mixed-integer linear programs are solved using the branch-and-bound solver in GLPK.

Simulation Results

For these results, the vehicle, trajectory, and failure cases are the same as given in Ref. 1. Presented in Figs. 1 and 2 are the time histories for the control moment error and the control effector command time histories. In Fig. 1, the moment error is defined by $\log_{10} \|\tilde{\boldsymbol{B}}\boldsymbol{\Lambda} - \boldsymbol{d}_{\text{des}}\|_2$. Recall that $\tilde{\boldsymbol{B}}$ is the piecewise linear analog to the $\boldsymbol{B}$ matrix that one encounters when using a linear approximation; therefore, the product $\tilde{\boldsymbol{B}}\boldsymbol{\Lambda}$ is the linear interpolation of the control moment data. We see that the modeling errors are negligible until the rudder failure is introduced at the 40 s mark. Between 40 and 60 s there is a control deficiency; therefore, the desired moment $\boldsymbol{d}_{\text{des}}$ is not feasible under the failure conditions. The performance of the mixed-optimization is identical to that given in Ref. 1 for the two-branch, piecewise linear control allocation problem.

The control surface time histories given in Fig. 2 are the control deflections returned by the control allocator. The control surface deflections compare favorably. There are very minor differences in the right flap deflection after the rudder failure is introduced, but this discrepancy does not appear to be an issue.

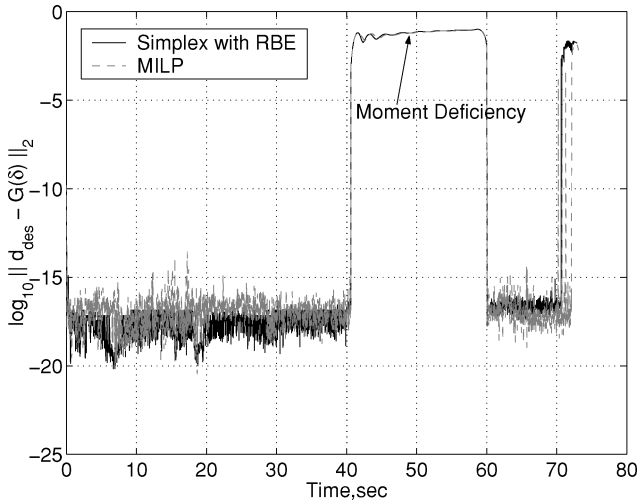

Fig. 1 Difference between commanded and applied moments.

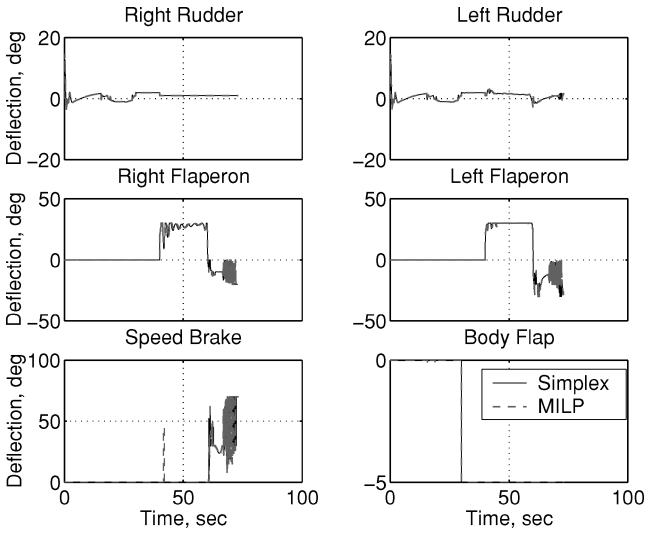

Fig. 2 Commanded control surface deflections.

Table 1 Solver execution time statistics: Athlon XP1800 + /Windows 2000

Method	Mean, s	Standard deviation, s	Maximum, s
MILP	0.0885	0.0423	0.4310
RBE	0.0083	0.0042	0.0700

Table 2 Mean and standard deviation of $\|\delta\|_2$

Method	Mean, deg	Standard deviation, deg
MILP	18.2365	8.6293
RBE	18.2657	8.5736

Table 3 Mean and standard deviation of $\|\tilde{\boldsymbol{B}}\boldsymbol{\Lambda} - \boldsymbol{d}_{\text{des}}\|_2$

Method	Mean	Standard deviation
MILP	4.4547×10^{-17}	1.0709×10^{-16}
RBE	1.4139×10^{-5}	2.6927×10^{-4}

Computational Performance Results

To compare execution time and solution accuracy for the mixed-integer linear program (MILP) and the linear programming formulation of the control allocation problem, we consider a fixed flight condition and 10,000 different control moment vectors. Each component of the control moment vector was selected randomly from a uniform distribution. For each control moment vector, the control allocation problem was solved using both an MILP solver and a simplex algorithm with restricted basis entry rules. The preference vector was taken to be $\boldsymbol{\delta}_p = \boldsymbol{0}$ for all cases. The execution time of both approaches was measured using the standard C library function `clock()` and is given in Table 1. It is apparent that the simplex method, on average, is an order of magnitude faster than the MILP approach. If it is assumed that the typical sample time of a digital flight-control system is 0.02 s, then the solution time given in Table 1 for the simplex method with restricted basis entry is more than adequate for practical application given a processor that is in the same class as that tested earlier. The mean control deflection for each method is given in Table 2. Note that for this particular set of random $\boldsymbol{d}_{\text{des}}$, the mean deflections are nearly the same for both approaches. On the other hand, Table 3 shows a significant difference in the average moment error. There were a small number of $\boldsymbol{d}_{\text{des}}$ for which the simplex method with restricted basis entry rules failed to find a set of control deflections that would produce a feasible moment, whereas the MILP formulation succeeded. The average error for these vectors is 0.4938 and is the cause for the poor correlation between the MILP solution and the simplex with restricted basis entry rules. For each of these vectors, the control deficiency occurs in the yawing moment. This deficiency occurs in about 0.5% of the cases that were tested. The cause of these deficiencies are related to an early termination of the simplex algorithm on a solution that is not a local optimum. In these cases, there are two restricted basis variables, $\lambda_i^{(k)}$ and $\lambda_i^{(k+1)}$, that are in the basis where one of the variables has a value of one and the other has value of zero due to roundoff error. In such a case, there may be a variable that has a negative cost coefficient, but, due to the restricted basis entry rules, is not eligible to enter the basis. Therefore, the algorithm terminates prematurely. This occurs when a vertex of the simplex lies exactly on a constraint line. Note, however, that if the early termination cases are removed, the simplex method with restricted basis entry rules has a mean error of 2.7381×10^{-17} and a standard deviation of 3.4196×10^{-17}.

Given that the control allocation problem, when solved using the modified simplex algorithm, was subject to early termination, it was hypothesized that the choice of $\boldsymbol{\delta}_p$ could have an effect on the solution returned by the control allocator. To study this hypothesis, we selected one of the cases that terminated early, randomly chose 1000 preference vectors, and resolved the control allocation problem. In this case, the simplex solver is sensitive to the selection

of $\boldsymbol{\delta}_p$ because approximately 28% of the cases terminated early. On the other hand, the MILP formulation that was solved by the branch-and-bound algorithm was insensitive to the selection of $\boldsymbol{\delta}_p$ because an optimal solution was always found. Unfortunately, there is no way to determine a priori whether the simplex algorithm will terminate early for a given $\boldsymbol{\delta}_p$ and $\boldsymbol{d}_{\text{des}}$. The same exercise was repeated for a moment vector that did not terminate early when solved using the simplex with restricted basis entry. In this case, both the simplex and the branch-and-bound solvers showed no sensitivity to the choice of $\boldsymbol{\delta}_p$. The question of whether there exists a means by which early termination of the simplex algorithm can be avoided remains an open problem.

Conclusions

The approach to nonlinear control allocation that was presented assumed that control moments generated by the deflection of aerodynamic surfaces were separable functions. This assumption allows one to approximate any separable nonlinear function by a piecewise linear function in the control allocation problem. As a result, we were able to cast a nonlinear optimization problem as a linear programming problem where a subset of the decision variables are subject to restricted basis entry rules. Although this still gives an approximate solution to the control allocation problem, it is much more accurate than the traditional methods that assume linear relationships between the control moments and the control effector positions. It was subsequently shown that a simplex algorithm that enforces the restricted basis entry rules can be solved fast enough for it to be a candidate for use in a real-time flight-control system, given a processor that is comparable to that available on today's desktop personal computers. However, the issue of early termination of the simplex algorithm must be addressed to ensure that when a feasible solution to the control allocation problem exists, it is always found.

Acknowledgment

This work was performed while the first author held a National Research Council Research Associateship Award at the U.S. Air Force Research Laboratory.

References

[1]Bolender, M. A., and Doman D. B., "Nonlinear Control Allocation Using Piecewise Linear Functions," *Journal of Guidance, Control, and Dynamics*, Vol. 27, No. 6, 2004, pp. 1017–1027.

[2]Bodson, M., "Evaluation of Optimization Methods for Control Allocation," *Journal of Guidance, Control, and Dynamics*, Vol. 25, No. 4, 2002, pp. 703–711.

[3]Buffington, J., "Modular Control Law Design for the Innovative Control Effectors (ICE) Tailless Fighter Aircraft Configuration 101-3," U.S. Air Force Research Lab., TR AFRL-VA-WP-TR 1999-3057, Wright–Patterson AFB, OH, June 1999.

[4]Bertsimas, D., and Tsitsiklis, J., *Introduction to Linear Optimization*, Athena Scientific, Belmont, MA, 1997, pp. 15–18 and 455, 456.

[5]Reklaitis, G., Ravindran, A., and Ragsdell, K., *Engineering Optimization: Methods and Application*, Wiley-Interscience, New York, 1983, pp. 314–324.

[6]Miller, C., "The Simplex Method for Local Separable Programming," *Recent Advances in Mathematical Programming*, edited by R. Graves and P. Wolfe, McGraw–Hall, New York, 1963, pp. 89–100.

[7]Hadley, G., *Nonlinear and Dynamic Programming*, Addison Wesley Longman, Reading, MA, 1964, pp. 104–134.