A Fuzzy Logic Approach to LQG Design With Variance Constraints

Emmanuel G. Collins, Jr., *Senior Member, IEEE*, and Majura F. Selekwa

***Abstract*—One of the well-known deficiencies of most modern control methods [i.e., H_2, H_∞, and L_1 (or ℓ_1) design] is that they attempt to represent multiple criteria with scalar cost functions. Hence, in practice the (static or dynamic) weights in the scalar cost function must be determined by an iterative process in order to satisfy the multiple objectives. It is of great time and cost benefit to automate this iterative process, but these problems tend to be highly nonlinear and extremely difficult to model analytically. However, a good designer can often observe trends and develop effective weight selection methodologies. The designer's logic is inherently "fuzzy" and it is hence natural to use fuzzy logic for algorithm implementation. This paper develops a fuzzy algorithm for selecting the weights in a linear quadratic Gaussian (LQG) cost functional such that constraints on the variances of the system are satisfied. This problem is denoted the variance constrained LQG (VCLQG) problem. Variations of this problem are considered in the existing literature using crisp logic. Numerical experiments show that when both the input and output variances are constrained, the fuzzy algorithm converges faster and tends to be much more robust to new systems or constraints than the crisp algorithms.**

***Index Terms*—Fuzzy logic, linear quadratic Gaussian (LQG) control, stochastic optimal control.**

I. INTRODUCTION

CONTROL methods based on "modern" control theory are based upon finding a control law that minimizes or constrains a scalar cost function. The use of optimization theories in the development of modern control laws leads to a degree of design automation, since after the cost function is chosen, the control law synthesis may be entirely performed by a computer-implemented numerical algorithm. However, as is well known, in most real-world problems the engineering objectives are multicriteria and cannot be easily captured by scalar cost criteria. Hence, in practice the control engineer must iteratively choose a set of weights, which may be static or dynamic, so that the scalar cost function actually yields a control law that satisfies the multiple objectives. So, despite greater utilization of the computer, modern control design has not yet reached a high degree of automation.

Because of the cost savings that automation affords, full automation of the control design process, including weight selection is highly desirable. The subject of weight selection in control design has attracted only very sporadic attention in the control literature (e.g., [6], [8], [11], [16], [17], [20], [22], and [25]). This is, perhaps, largely due to the fact that these problems are highly nonlinear, and it is in general very difficult to analytically model the relationships between the cost function weights and the various design objectives.

Fortunately, there often are observable relationships between the various cost function weights and the multiple criteria. In this case, with experience, a modern control designer can develop methodologies to pick the weights to satisfy the multiple criteria. Hence, it is possible to automate the design process by capturing the designer's observations and experience and implementing the resulting design rules in iterative computer algorithms. This implementation can be attempted by using crisp logic. However, since the designer's thinking and design principles are in reality "fuzzy," it is more natural to use fuzzy logic. This paper focuses on the use of fuzzy logic to choose weights in a weight selection problem that has received substantial attention in the literature.

The problem considered is variance constrained linear quadratic Gaussian (VCLQG) design. The basic VCLQG problem is to pick the weights in an LQG (i.e., H_2) cost function so that the variances of the system inputs and outputs are constrained by selected amounts. A variant of this problem is considered in [16], where the task of minimizing a preselected quadratic cost function subject to input and output variance constraints is considered. In other references [20], [25] the dual problems considered are: 1) minimize a weighted sum of the differences between each input variance and its constraint subject to constraints on the output variances [the output variance assignment (OVA) problem] or 2) minimize a weighted sum of the difference between each output variance and its constraint subject to constraints on the input variances [the input variance assignment (IVA) problem]. The output covariance control (OCC) problem considered in [11] seeks to minimize an input cost subject to constraints on the output covariances. The OCC problem can be reduced to that of minimizing the input cost subject to constraints on output variances. Here, we consider the most general VCLQG problem and report a fuzzy weight selection methodology. It should be mentioned that variance constraints also have a deterministic interpretation since the variances bound the infinity norms of the inputs and outputs when the system is disturbed by a finite energy signal. This interpretation has practical value and the reader is referred to [11] and [25] for more details.

All of the previous algorithms for variations of the VCLQG design problem are based on crisp logic. The algorithm in [16] is

Manuscript received February 28, 2000. Manuscript received in final form March 20, 2001. Recommended by Associate Editor L. K. Mestha. This work was supported in part by the National Science Foundation under Grant CMS-9802197.

The authors are with the Department of Mechanical Engineering, Florida A&M University, Florida State University College of Engineering, Tallahassee, FL 32310 USA (e-mail: ecollins@eng.fsu.edu; majura@eng.fsu.edu).

Publisher Item Identifier S 1063-6536(02)00328-7.

a nonlinear programming (NLP) quasi-Newton algorithm while the algorithms in [20] for the OVA and IVA problems are heuristically derived. The output variance control (OVC) and the input variance control (IVC) algorithms in [11] are variants of the OVA and IVA algorithms, respectively. The algorithm in [25] for the output covariance control (OCC) problem is based on the theory of convergent sequences in a closed convex set, and is provably convergent. The primary motivation for the developments here is to consider the most general form of VCLQG and to demonstrate that fuzzy methodologies can be developed for multicriteria problems. It is seen that the fuzzy algorithm tends to converge faster and be much more numerically robust than the crisp algorithms. However, an ultimate advantage of fuzzy weight selection methodologies may be their extendability to other multicriteria problems.

Note that the use of fuzzy logic here differs from most usages of fuzzy logic in control. In most control applications (see, for example, [4], [5], [18], [19], [23], and [24]) fuzzy logic is used as a replacement for conventional control laws. In contrast, here fuzzy logic is used to aid conventional control design methods.

This paper is organized as follows. Section II describes and formulates the VCLQG problem. It then describes several variants of the VCLQG problem along with the crisp algorithms proposed for their solutions. Section III develops a fuzzy approach to weight selection in VCLQG design. Section IV presents numerical examples, and Section V discusses the conclusions.

Notation: $\mathbb{R}$ is the field of real numbers, $\mathrm{tr}(M)$ is the trace of matrix M, I_r is an $r \times r$ identity matrix, O_r is an $r \times r$ zero matrix, $\mathrm{row}_i(M)$ represents the ith row of matrix M, $\mathrm{col}_i(M)$ represents the ith column of matrix M, $\mathbf{E}$ is the expectation operator and $\|v\|_\infty$ is the infinity norm of vector v, i.e., $=\max_i |v_i|$.

II. THE VCLQG PROBLEM AND ITS VARIANTS

This section begins by describing the VCLQG problem. Subsequently, several variants of the VCLQG [11], [16], [20], [25] problem are described along with the algorithms that have been proposed to solve them.

A. The VCLQG Problem

Consider a linear time-invariant plant

$$\dot{x}(t) = Ax(t) + Bu(t) + D_1 w(t) \tag{1}$$

$$y(t) = Cx(t) + D_2 w(t) \tag{2}$$

$$z(t) = Ex(t) \tag{3}$$

where $x \in \mathbb{R}^n$ is the state vector, $u \in \mathbb{R}^m$ is the control input vector, $y \in \mathbb{R}^p$ is the measurement vector, $z \in \mathbb{R}^q$ is the output performance vector, w is a white noise vector process with zero mean and identity covariance, (A, B) is stabilizable, (A, C) is detectable, D_2 is full column rank, and $D_1 D_2^T = 0$. The standard LQG control problem is that of finding a control input u that minimizes the quadratic performance function

$$J = \mathrm{tr}(Q_x V_x) + \mathrm{tr}(Q_u V_u) \tag{4}$$

where

$$Q_x = Q_x^T \geq 0, \qquad Q_u = Q_u^T > 0 \tag{5}$$

and V_x and V_u denote, respectively, the steady-state covariance matrices of the state and input vectors, i.e.,

$$V_x = \lim_{t\to\infty} \mathbf{E}\left[x(t)x^T(t)\right], \quad V_u = \lim_{t\to\infty} \mathbf{E}\left[u(t)u^T(t)\right]. \tag{6}$$

The weighting matrices Q_x and Q_u are the LQG weights which are to be selected by the control designer to express the relative importance of the various states (or outputs) and control inputs.

The solution to the LQG control problem is well established (see, for example, [1]) and results in a controller whose state-space description is

$$\dot{x}_c(t) = A_c x_c(t) + B_c y(t) \tag{7}$$

$$u(t) = -C_c x_c(t) \tag{8}$$

where

$$A_c = A - B_c C + BC_c \tag{9}$$

$$B_c = QC^T V_2^{-1}, \qquad C_c = Q_u^{-1} B^T P. \tag{10}$$

P and Q are the respective solutions of the regulator and observer Ricatti equations

$$0 = A^T P + PA + Q_x - PBQ_u^{-1} B^T P \tag{11}$$

$$0 = AQ + QA^T + V_1 - QC^T V_2^{-1} CQ \tag{12}$$

and V_1 and V_2 are defined as

$$V_1 \triangleq D_1 D_1^T, \qquad V_2 \triangleq D_2 D_2^T. \tag{13}$$

It has also been shown [12] that for the LQG controller the state and input covariance matrices which minimize the quadratic performance function (4) are, respectively

$$V_x = Q + \hat{Q}, \qquad V_u = C_c \hat{Q} C_c^T \tag{14}$$

where $\hat{Q}$ is the solution of the Lyapunov equation

$$0 = (A - BC_c)\hat{Q} + \hat{Q}(A - BC_c)^T + B_c V_2 B_c^T. \tag{15}$$

The covariance matrices V_x and V_u (and hence the variances) are highly dependent on the weighting matrices Q_x and Q_u since they determine the solution of the regulator Ricatti equation.

The Variance Constrained LQG (VCLQG) Problem: Find the weighting matrices Q_x and Q_u such that the LQG control law yields a closed-loop system satisfying

$$\lim_{t\to\infty} \mathbf{E}\left[z_i^2(t)\right] = \mathrm{tr}(Q_i V_x) \leq \sigma_i^2, \qquad i = 1, 2, \ldots, q \tag{16}$$

$$\lim_{t\to\infty} \mathbf{E}\left[u_j^2(t)\right] = \mathrm{tr}(R_j V_u) \leq \mu_j^2, \qquad j = 1, 2, \ldots, m \tag{17}$$

where

$$Q_i = \mathrm{row}_i(E)\mathrm{col}_i(E^T), \quad R_j = \mathrm{row}_j(I_m)\mathrm{col}_j(I_m^T). \tag{18}$$

To formulate a solution to the VCLQG problem, let

$$Q_x = E^T Q_z E, \qquad Q_z \geq 0 \tag{19}$$

and express the cost function (4) as

$$J = \mathrm{tr}(Q_z V_z) + \mathrm{tr}(Q_u V_u) \tag{20}$$

where V_z is the steady-state covariance matrix of the output performance vector $z(t)$, i.e.,

$$V_z = \lim_{t\to\infty} \mathbf{E}\left[z(t)z^T(t)\right]. \tag{21}$$

Furthermore, adopting some results from [7], [11], [13]–[15], [21], and [22] we restrict the weighting matrices Q_z and Q_u to be diagonal such that

$$\begin{aligned} Q_z &= \mathrm{diag}(\alpha_{z,1}, \ldots, \alpha_{z,q}) \\ Q_u &= \mathrm{diag}(\alpha_{u,1}, \ldots, \alpha_{u,m}). \end{aligned} \tag{22}$$

Q_x and Q_u can now be expressed as

$$Q_x = \sum_{i=1}^{q} \alpha_{z,i} Q_i, \qquad Q_u = \sum_{j=1}^{m} \alpha_{u,j} R_j \tag{23}$$

indicating that the LQG cost function is a weighted sum of the steady-state variances of the inputs and performance outputs. Let the set $\{\alpha_i\}_{i=1}^{q+m}$ be defined such that

$$\alpha_i \triangleq \begin{cases} \alpha_{z,i}, & i = 1, 2, \ldots, q \\ \alpha_{u,i-q}, & i = q+1, q+2, \ldots, q+m. \end{cases} \tag{24}$$

The VCLQG problem reduces to that of determining the parameter set $\{\alpha_i\}_{i=1}^{q+m}$ which from now on will be called the weights. The weights $\{\alpha_i\}_{i=1}^{q+m}$ which solve the VCLQG problem for a particular set of variance constraints (16) and (17) are not unique, since all scaled sets the solution are also solutions. Depending on the solution methodology, the VCLQG problem may not be easy to solve.

B. The Optimal VCLQG Problem

A variant of the VCLQG problem is the optimal VCLQG problem [2], [16], which optimizes the fixed quadratic cost function (4), where $Q_x = Q_x^T \geq 0$ and $Q_u = Q_u^T \geq 0$, subject to the inequality constraints (16) and (17). In [2] and [16] this problem is solved by finding the nonnegative Lagrange multipliers $\{\lambda_{z,i}\}_{i=1}^{q}$ and $\{\lambda_{u,j}\}_{j=1}^{m}$ such that the control law that minimizes the Lagrangian function

$$J_L = \mathrm{tr}\, Q_x^* V_x + \mathrm{tr}\, Q_u^* V_u \tag{25}$$

where

$$Q_x^* = Q_x + \sum_{i=1}^{q} \lambda_{z,i} Q_i, \quad Q_u^* = Q_u + \sum_{j=1}^{m} \lambda_{u,j} R_j \tag{26}$$

also satisfies the inequality constraints (16) and (17) and the complementary slackness condition

$$\sum_{i=1}^{q} \lambda_{z,i}\left[\mathrm{tr}(Q_i V_x) - \sigma_i^2\right] + \sum_{j=1}^{m} \lambda_{u,j}\left[\mathrm{tr}(R_j V_u) - \mu_j^2\right] = 0. \tag{27}$$

In [2], the Lagrange multipliers are scaled such that

$$\lambda_1 + \lambda_2 + \cdots + \lambda_{q+m} = 1 \tag{28}$$

where the set $\{\lambda_i\}_{i=1}^{q+m}$ is defined such that

$$\lambda_i \triangleq \begin{cases} \lambda_{z,i}, & i = 1, 2, \ldots, q \\ \lambda_{u,i-q}, & i = q+1, q+2, \ldots, q+m. \end{cases} \tag{29}$$

In [16], Q_x and Q_u are assumed known such that the optimal VCLQG problem may be viewed as that of finding the weights Q_x^* and Q_u^* in the cost function (25) so that the resulting control law satisfies the inequality constraints (16) and (17). The set of the Lagrange multipliers $\{\lambda_i\}_{i=1}^{q+m}$ which solve this problem is determined iteratively from some initial values $\{\lambda_i^{(0)}\}_{i=1}^{q+m}$ using the modified quasi-Newton update in which each element λ_i is updated as [16]

$$\lambda_i^{(k+1)} = \lambda_i^{(k)} + \mathrm{sat}\left(\beta^{(k)} H_i^{(k)} \xi_i\left(\lambda_i^{(k)}\right), \overline{\sigma}\lambda_i^{(k)}\right) \tag{30}$$

where $\overline{\sigma} \in (0, 1)$ is selected by the user, the matrix H_i is the quasi-Newton approximation of the inverse of the Jacobian matrix of $\xi\colon \mathbb{R}^{q+m} \to \mathbb{R}^{q+m}$ with respect to λ_i, where the elements of ξ are defined by

$$\xi_i = \begin{cases} \lambda_i\left[\mathrm{tr}(Q_i V_x) - \sigma_i^2\right], & i = 1, 2, \ldots, q \\ \lambda_i\left[\mathrm{tr}(R_{i-q} V_u - \mu_{i-q}^2\right], & i = q+1, q+2, \ldots, q+m. \end{cases} \tag{31}$$

The saturation function sat $(\cdot, \cdot)$ in (30) is defined as

$$\mathrm{sat}(x, y) = \begin{cases} x, & \text{if } |x| \leq |y| \\ \mathrm{sgn}(x) \times |y|, & \text{otherwise} \end{cases} \tag{32}$$

and $\beta^{(k)}$ is a positive parameter updated from some initial value $\beta^{(0)}$ according to

$$\beta^{(k+1)} = \beta^{(k)}\left(\gamma - \beta^{(k)}\right)(\gamma - 1)^{-1} \tag{33}$$

where γ is a positive number selected by the user. The convergence rate of the above algorithm, called here the nonlinear programming (NLP) algorithm, depends on the user selected parameters $\overline{\sigma}$, $\beta^{(0)}$, and γ. The optimal values of these parameters depend on both the system (1)–(3) and the constraints (16), (17). When optimal values for these parameters are used, the NLP algorithm converges at a quadratic rate. For further details about the NLP algorithm, the reader is referred to [16].

Note that if in the optimal VCLQG problem weighting matrices Q_x and Q_u that determine the fixed quadratic cost function (4) are chosen to be zero, then

$$Q_x^* = \sum_{i=1}^{q} \lambda_i Q_i, \qquad Q_u^* = \sum_{j=1}^{m} \lambda_{q+j} R_j \tag{34}$$

which have the same respective forms as (23). In this case the optimal VCLQG problem reduces to the VCLQG problem.

In [2], the Lagrange multipliers that solve the optimal VCLQG problem are determined by using the constrained cutting plane and ellipsoid algorithms. These algorithms use the value of the cost function and its subgradients to shift the upper and lower bounds on the weights until they converge to the desired weights. Since the algorithms work by shrinking the zone that contains the desired solution, proper initialization is required. The initial zone must contain the desired solution outright. The initialization of the constrained cutting plane and ellipsoid algorithms is done using the Bryson's rule [3]. It is noted that these algorithms resemble in many respects to the NLP algorithm, and were not used for comparison in this paper.

C. The OVA Problem and the IVA Problem

The OVA and the IVA are considered in [20]. In [11] the OVA and the IVA problems are generalized to the IVC and the OVC problems. However, these problems and the proposed algorithms for their solution are very similar. Therefore, here we restrict our discussion to the IVA and the OVA problems.

The OVA problem is that of assigning the weights $\{\alpha_i\}_{i=1}^{q+m}$ such that the binding output variance constraints (16) are taken to their boundaries and the weighted sum of the errors on the relaxed input variance constraints (17) is minimized, i.e.,

$$\min_{\{\alpha_i\}} \sum_{i=q+1}^{q+m} \left(\lim_{t\to\infty} \mathbf{E}\left[u_{i-q}^2(t)\right] - \mu_{i-q}^2 \right) \tag{35}$$

subject to

$$\lim_{t\to\infty} \mathbf{E}\left[z_i^2(t)\right] - \sigma_i^2 = 0, \qquad i = 1,\, 2,\, \ldots,\, q. \tag{36}$$

In [20] the output and input weights $\{\alpha_{z,\,i}\}_{i=1}^{q}$ and $\{\alpha_{u,\,j}\}_{j=1}^{m}$ are determined iteratively by the OVA algorithm from some initial values according to

$$\alpha_{z,\,i}^{(k)} = \frac{\lim_{t\to\infty} \mathbf{E}\left[z_i^2(t)\right]}{\sigma_i^2}\, \alpha_{z,\,i}^{(k-1)} \tag{37}$$

and

$$\alpha_{u,\,j}^{(k)} = \begin{cases} \dfrac{\lim_{t\to\infty} \mathbf{E}\left[u_j^2(t)\right]}{\mu_j^2}\, \alpha_{u,\,j}^{(k-1)}, & \text{if } \dfrac{\lim_{t\to\infty} \mathbf{E}\left[u_j^2(t)\right]}{\mu_j^2} \le 1 \\ \alpha_{u,\,j}^{(k-1)}, & \text{otherwise.} \end{cases} \tag{38}$$

Basically, the weight updates used by the OVA algorithm assume that the cost index (36) can be minimized and each output variance can be taken to the constraint boundary in a single iteration. When this assumption is not satisfied, the algorithm iterates until convergence.

The IVA problem, dual to the OVA problem, is that of assigning the weights such that the binding input variance constraints (17) are taken to their boundaries and weighted sum of the errors on the relaxed output variance constraints (16) is minimized, i.e.,

$$\min_{\{\alpha_i\}} \sum_{i=1}^{q} \left(\lim_{t\to\infty} \mathbf{E}\left[z_i^2(t)\right] - \sigma_i^2 \right) \tag{39}$$

subject to

$$\lim_{t\to\infty} \mathbf{E}\left[u_{i-q}^2(t)\right] - \mu_{i-q}^2 = 0, \qquad i = q+1, q+2, \ldots, q+m. \tag{40}$$

Likewise, the IVA algorithm is the dual of the OVA algorithm.

D. The OCC Problem

The OCC problem [25] is that of finding a static or dynamic feedback controller that minimizes the input performance cost

$$J_u = \lim_{t\to\infty} \mathbf{E}\left[u(t)^T Q_u u(t)\right] = \mathrm{tr} Q_u V_u, \qquad Q_u > 0 \tag{41}$$

subject to constraints on the output covariances

$$\lim_{t\to\infty} \mathbf{E}\left[z_i(t) z_i(t)^T\right] = E_i V_x E_i^T \le \overline{\Sigma}_i, \qquad i = 1, 2, \ldots, p \tag{42}$$

where $\overline{\Sigma}_i > 0$, $i = 1,\, 2,\, \ldots,\, p$, and $z(t)$ and E have the compatibly partitioned forms

$$\begin{aligned} z(t) &= \left[z_1(t)^T,\, \ldots,\, z_p(t)^T\right]^T \\ E &= \left[E_1^T,\, \ldots,\, E_p^T\right]^T. \end{aligned} \tag{43}$$

The input weight Q_u is assumed to be known. If it is assumed that $p = q$ such that each $z_i(t)$ is a scalar as in the VCLQG problem, then each E_i will be a row vector and (42) becomes equivalent to (16). Notice that this problem does not explicitly consider constraints on the individual input variances.

It is shown in [25] that the optimal solution for this problem exists only if there is a symmetric positive semidefinite matrix Q_z defined as in (22) such that

$$\sum_{i=1}^{q} \alpha_{z,i}\left(E_i V_x E_i^T - \sigma_i^2\right) = 0 \qquad \text{where } \mathrm{tr}(Q_i V_x) \le \sigma_i^2. \tag{44}$$

To establish the optimal solution to the OCC problem, one must ensure that Q_z exists. It follows that the OCC problem may be viewed as that of finding diagonal Q_z satisfying (44) such that for known Q_u, the control law which minimizes cost function

$$J = \mathrm{tr}(Q_u V_u) + \mathrm{tr}(Q_z[V_z - V_c]) \tag{45}$$

where $V_c = \mathrm{diag}(\sigma_1^2,\, \ldots,\, \sigma_q^2)$, also satisfies (16).

The OCC algorithm which solves the OCC problem is described in detail in [25]. If $p = q$, each diagonal element $\alpha_{z,\,i}$ of Q_z is updated iteratively from some initial value $\alpha_{z,\,i}^{(0)}$ according to

$$\begin{aligned} \alpha_{z,\,i}^{(k+1)} &= \beta \alpha_{z,\,i}^{(k)} + (1-\beta) \times \\ &\qquad \mathcal{P}\left[\alpha_{z,\,i}^{(k)} + \overline{\gamma}\left(\lim_{t\to\infty} \mathbf{E}\left[z_i(t)^2\right] - \sigma_i^2\right)\right] \end{aligned} \tag{46}$$

until (44) is satisfied, giving $\alpha_{z,\,i}^{(k+1)} = \alpha_{z,\,i}^{(k)}$. Here, the function $\mathcal{P}[M]$ is defined as

$$\mathcal{P}[M] = \begin{cases} 0, & \text{for } M \le 0 \\ M & \text{otherwise.} \end{cases} \tag{47}$$

The nonnegative parameters $\overline{\gamma} > 0$ and $\beta \in (0,\, 1)$ must be selected by the user and the values which optimize the convergence rate are dependent upon both the system (1)–(3) and the constraints (16), (17).

III. FUZZY ALGORITHM FOR QUADRATIC WEIGHT SELECTION

A. Solution Approach

To develop a fuzzy algorithm for selection of the weights $\{\alpha_i\}_{i=1}^{q+m}$, we introduce the following variables. Let $\{v_i\}_{i=1}^{q+m}$, denote the set of input and output variances, i.e.,

$$v_i \triangleq \begin{cases} \lim\limits_{t\to\infty} \mathbf{E}\big[z_i^2(t)\big], & i = 1, 2, \ldots, q \\ \lim\limits_{t\to\infty} \mathbf{E}\big[u_{i-q}^2(t)\big], & i = q+1, q+2, \ldots, q+m \end{cases} \tag{48}$$

and the set of LQG weights $\{\alpha_i\}_{i=1}^{q+m}$ be defined as in (24). The set of input and output variances corresponding to unity weights (i.e., $\alpha_i = 1$, $i = 1, 2, \ldots, q+m$) is denoted by $\{\overline{v}_i\}_{i=1}^{q+m}$, and the corresponding set of variance constraints is denoted by $\{\zeta_i\}_{i=1}^{q+m}$, where

$$\zeta_i \triangleq \begin{cases} \sigma_i^2, & i = 1, 2, \ldots, q \\ \mu_{i-q}^2, & i = q+1, q+2, \ldots, q+m. \end{cases} \tag{49}$$

The deviation of the variance v_i from its constraint ζ_i, normalized about the corresponding unity weight variance $\overline{v}_i$ is denoted by ε_i, i.e.,

$$\varepsilon_i \triangleq \frac{v_i - \zeta_i}{\overline{v}_i}, \qquad i = 1, 2, \ldots, q+m. \tag{50}$$

Since each element of the set $\{\varepsilon_i\}_{i=1}^{q+m}$ depends on each element of the LQG weight set $\{\alpha_i\}_{i=1}^{q+m}$, we formulate the VCLQG problem as that of finding a set $\{\alpha_i\}_{i=1}^{q+m}$ which satisfies the nonlinear inequalities

$$\varepsilon_i(\alpha_1, \alpha_2, \ldots, \alpha_{q+m}) \leq 0, \qquad i = 1, 2, \ldots, q+m. \tag{51}$$

The solution for such problem is done iteratively from some initial estimate $\alpha_i^{(0)}$ using

$$\alpha_i^{(k+1)} = \alpha_i^{(k)} + \Delta\alpha_i^{(k)}. \tag{52}$$

Note that there are many sets $\{\alpha_i\}_{i=1}^{q+m}$ which satisfy (51), however, we seek the set which takes ε_i close to zero. In practice, a human designer solves this problem by adjusting $\Delta\alpha^{(k)}$ according to observed patterns until satisfactory performance is achieved.

B. The Fuzzy Logic for VCLQG Weight Selection

The value of each input or output variance v_i is determined by the entire set of LQG weights $\{\alpha_i\}_{i=1}^{q+m}$. However, the sign of the change in v_i due to a change in α_i is known *a priori*, whereas the sign of the change in v_i due to a change in α_j $(j \neq i)$ is *not* known *a priori*. In particular, due to optimality it is known that v_i will decrease with increasing α_i and increase with decreasing α_i, while for $j \neq i$ optimality does not predict whether v_i will increase or decrease as α_j changes. These trends are illustrated by Figs. 1 and 2 which correspond to a system with two inputs (u_1, u_2) and two performance outputs (z_1, z_2). It is seen that while the variance of input 2 decreases monotonically as the corresponding weight $\alpha_{u,2}$ is increased as shown by the solid curve in Fig. 1, depending on the choice of the remaining weights, the variances of input 1 and the two outputs may not even vary monotonically, as shown clearly by the solid curve in Fig. 2.

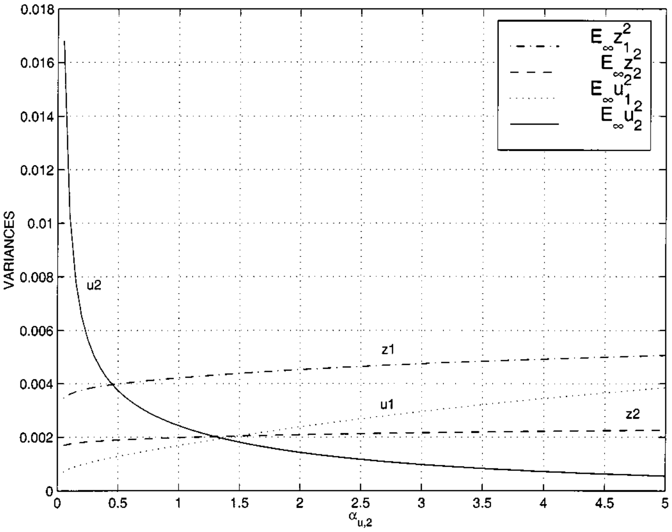


Fig. 1. Variation of the input and output variances with the weight $(\alpha_{u,2})$ on input variance 2 $(\alpha_{u,1} = \alpha_{z,1} = \alpha_{z,2} = 1)$.

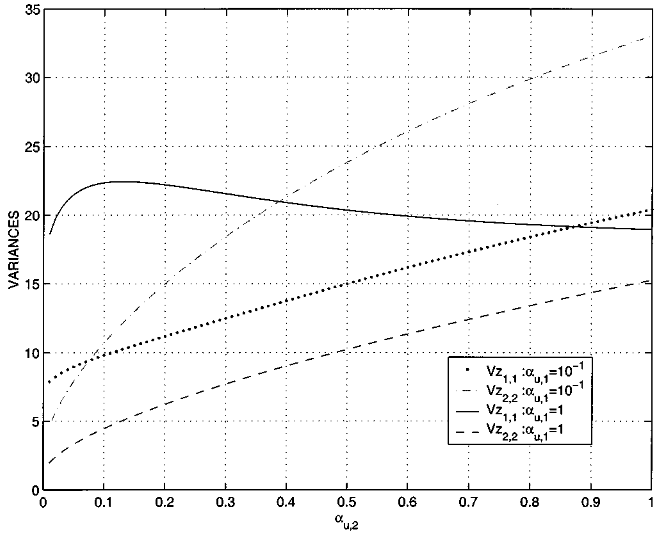


Fig. 2. Variation of the output variances with the weight $\alpha_{u,2}$ on input variance 2 for two values of $\alpha_{u,1}$ $(\alpha_{z,1} = \alpha_{z,2} = 1)$.

Since the sensitivity of the variance to the weight changes is not constant, the weight adjustments should be based not only on the deviation of the variances from their design constraints, but also the *sensitivity* of the variances with respect to each weight. The sensitivities are denoted by η_{ij} and are precisely defined by

$$\eta_{ij} \triangleq \frac{\partial v_i}{\partial \alpha_j} = \overline{v}_i \frac{\partial \varepsilon_i}{\partial \alpha_j}, \qquad i, j = 1, 2, \ldots, q+m \tag{53}$$

corresponding to the gradient of the v_i versus α_j curve. In practice we cannot compute η_{ij} precisely. Hence we use an estimate $\tilde{\eta}_{ij}$ given by

$$\tilde{\eta}_{ij} = \overline{v}_i \frac{\delta \varepsilon_i}{\delta \alpha_j} \approx \eta_{ij} \tag{54}$$

where $\delta\alpha_j$ denotes a small change in α_j and $\delta\varepsilon_i$ denotes the corresponding change in ε_i. In the fuzzy algorithm we use the normalized sensitivity estimate $\overline{\eta}_{ij}$ defined as

$$\overline{\eta}_{ij} \triangleq \frac{\tilde{\eta}_{ij}}{\overline{v}_i} = \frac{\delta\varepsilon_i}{\delta\alpha_j} \tag{55}$$

corresponding to the gradient of the ε_i versus α_j curve.

Let $\{\alpha_i^{(k)}\}_{i=1}^{q+m}$ denote the present set of LQG weights which yields the set of normalized variance deviations $\{\varepsilon_i^{(k)}\}_{i=1}^{q+m}$. Assume for the moment that a weight α_i does not affect any of the variances v_j for $j \neq i$ (i.e., the v_j versus α_i curves are horizontal) and the v_i versus α_i curve is linear. Then, each variance constraint would equal its bound ($\varepsilon_i = 0$) if $\alpha_i \leftarrow \alpha_i^{(k)} + \Delta\alpha_i^{(k)}$, $i = 1, 2, \ldots, q+m$, where

$$\Delta\alpha_i^{(k)} = \frac{\varepsilon_i^{(k)}}{\overline{\eta}_{ii}^{(k)}}, \qquad i = 1, 2, \ldots, q+m. \tag{56}$$

However, if there exists $j\,(j \neq i)$ such that v_j is sensitive to changes in α_i and the v_j versus α_i curve is also linear, then its normalized variance error ε_j will change by

$$\Delta\varepsilon_{j(i)}^{(k)} = \overline{\eta}_{ji}^{(k)} \Delta\alpha_i^{(k)}. \tag{57}$$

This simplified analysis illustrates that the weights α_i and α_j will need to be readjusted further to ensure that ε_j remains (approximately) unchanged.

Below we describe qualitatively an experimentally derived algorithm for finding a set of LQG weights $\{\alpha_i\}_{i=1}^{q+m}$ that satisfy the variance constraints (16) and (17). The general flow of the algorithm is as follows:

1) Choose an initial set of LQG weights.
2) For $i = 1, 2, \ldots, q+m$
— Use the normalized variance deviations and approximations of the sensitivities to choose a new set of weights such that ε_i is approximately zero while ε_j is approximately unchanged ($j \neq i$).
3) Repeat the previous step until convergence.

In the following, the algorithm is described in greater detail using fuzzy language where the sizes of the variables are described as being "small," "large," etc., relative to some values determined based on the present maximum error. In particular, the absolute value ($|\varepsilon_i|$) of a normalized variance deviation is described relative to the size of the largest error present $\overline{\varepsilon}_{\max}$ defined by

$$\overline{\varepsilon}_{\max} \triangleq \max_i |\varepsilon_i|. \tag{58}$$

The absolute values ($|\Delta\alpha_i|$) of the variation in an LQG weight is described relative to the weight which would be required to bring $\overline{\varepsilon}_{\max}$ to zero under the assumption that the v_i versus α_i curve is linear. In particular, if

$$\bar{i} = \arg\max_i |\varepsilon_i| \tag{59}$$

then using (56) the variation size $|\Delta\alpha_i|$ is described relative to $\Delta\overline{\alpha}_{\max}$ defined by

$$\Delta\overline{\alpha}_{\max} \triangleq \left| \frac{\varepsilon_{\bar{i}}}{\overline{\eta}_{\bar{i}\bar{i}}} \right|. \tag{60}$$

The size of the absolute value $|\overline{\eta}_{ji}|$ of a normalized sensitivity estimate is described relative to the interval [0, 1].

Fuzzy Algorithm for VCLQG Weight Selection

Step 1. Set all weights to unity, i.e., $\alpha_i = 1, i = 1, 2, \ldots, q+m$ and compute the set of unity weight variances $\{\overline{v}_i\}_{i=1}^{q+m}$.

Step 2. Choose an initial set of LQG weights $\{\alpha_{i,0}\}_{i=1}^{q+m}$ and initialize $\alpha_i = \alpha_{i,0}$, $i = 1, 2, \ldots q+m$.

Step 3. For $i = 1, 2, \ldots, q+m$

3a. Compute the set of normalized variance deviations $\{\varepsilon_i\}_{i=1}^{q+m}$ and the set of normalized sensitivity estimates $\{\overline{\eta}_{ij}\}_{i,j=1}^{q+m}$. Let $\varepsilon_{i,0} = \varepsilon_i$, $i = 1, 2, \ldots, q+m$.

3b. Use $\varepsilon_i, \overline{\eta}_{ii}$ and fuzzy logic of the following type to choose a value of $|\Delta\alpha_i|$ such that [with the sign of $\Delta\alpha_i$ determined by (61) below] $\alpha_i \leftarrow \alpha_i + \Delta\alpha_i$ yields $\varepsilon_i \approx 0$: 1) if $|\varepsilon_i|$ is very large and $|\overline{\eta}_{ii}|$ is small then choose $|\Delta\alpha_i|$ very large; 2) if $|\varepsilon_i|$ is medium and $|\overline{\eta}_{ii}|$ is medium then choose $|\Delta\alpha_i|$ medium; and 3) if $|\varepsilon_i|$ is small and $|\overline{\eta}_{ii}|$ is small then choose $|\Delta\alpha_i|$ small; etc. (The full logic is shown below in Table IV.)

3c. For $j = 1, 2, \ldots, q+m\ (j \neq i)$

i. Use $\Delta\alpha_i$, η_{ji} and fuzzy logic of the following type to estimate $|\Delta\varepsilon_{j(i)}|$ such that [with the sign of $\Delta\varepsilon_{j(i)}$ determined by Table I or equivalently (62) below] $\Delta\varepsilon_{j(i)}$ is an estimate of the change in ε_j due to the weight change $\Delta\alpha_i$: 1) if $|\Delta\alpha_i|$ is large and $|\overline{\eta}_{ij}|$ is large then $|\Delta\varepsilon_{j(i)}|$ is large; 2) if $|\Delta\alpha_i|$ is medium and $|\overline{\eta}_{ij}|$ is medium then $|\Delta\varepsilon_{j(i)}|$ is medium; 3) if $|\Delta\alpha_i|$ is small and $|\overline{\eta}_{ij}|$ is small then $|\Delta\varepsilon_{j(i)}|$ is small; etc. (The full logic is shown below in Table V.)

ii. Use $\Delta\varepsilon_{j(i)}$, $\overline{\eta}_{jj}$ and the fuzzy logic of Step 3b to choose $\Delta\alpha_{j(i)}$ such that $\alpha_j \leftarrow \Delta\alpha_{j(i)} + \alpha_j$ yields $\varepsilon_j \approx \varepsilon_{j,0}$ (i.e., the influence of $\Delta\alpha_i$ on ε_j is negated).

iii. Use $\Delta\alpha_{j(i)}, \overline{\eta}_{ji}$, and the fuzzy logic of Step 3c-i to choose $\Delta\varepsilon_{i(j(i))}$, an estimate of the change in ε_i due the weight change $\Delta\alpha_{j(i)}$.

iv. Use $\Delta\varepsilon_{i(j(i))}, \overline{\eta}_{ii}$, and the fuzzy logic of Step 3b to choose $\Delta\alpha_{i(j(i))}$ such that $\varepsilon_i \approx 0$ (i.e., the influence of $\Delta\alpha_j$ on ε_i is negated).

3d. Update the weight α_i using $\alpha_i \leftarrow \alpha_i + \Delta\alpha_i^*$ where $\Delta\alpha_i^* = (\Delta\alpha_i + \sum_j \Delta\alpha_{i(j(i))})$. Then, adjust the weights to ensure that the definiteness constraints (5) are satisfied (as described below in Section III-D).

Step 4. Repeat Step 3 until convergence. (See Section III-D for details on the convergence criteria.)

Although in this algorithm it is assumed that all inputs and outputs are constrained, it is straight forward to select and constrain only specific channels such as the inputs, outputs, or some of the inputs and outputs. Details of the algorithm are given in the remainder of this section.

TABLE I
SIGN RELATIONSHIP BETWEEN INPUTS AND OUTPUTS OF THE WC FIS

Inputs		Output
$\Delta\alpha_i$	$\bar{\eta}_{ji}$	$\Delta\varepsilon_{j(i)}$
+	+	+
+	-	-
-	+	-
-	-	+

TABLE II
DEFINITIONS OF THE UDs FOR THE DIFFERENT FIS VARIABLES

Variable	UD Range	Definitions
$\|\varepsilon_*\|$	$[0, \bar{\varepsilon}_{\max}]$	$\bar{\varepsilon}_{\max} = \max_i \varepsilon_i^{(k)}$
$\|\Delta\alpha_*\|$	$[0, \Delta\bar{\alpha}_{\max}]$	$\Delta\bar{\alpha}_{\max} = \frac{\varepsilon_{\bar{i}}}{\eta_{\bar{i}\bar{i}}} : \bar{i} = \arg\max_i \varepsilon_i^{(k)}$
$\|\eta_{ij}\|$	[0,1]	gradient of $\alpha_j - \varepsilon_i$ curve at 45°

TABLE III
FUZZY MEMBERSHIP GROUPS FOR THE VARIABLES

Group	Linguistic Label	Membership function
1	Small (S)	Triangular
2	Medium (M)	Triangular
3	Large (L)	Triangular
4	Very Large (VL)	s-function

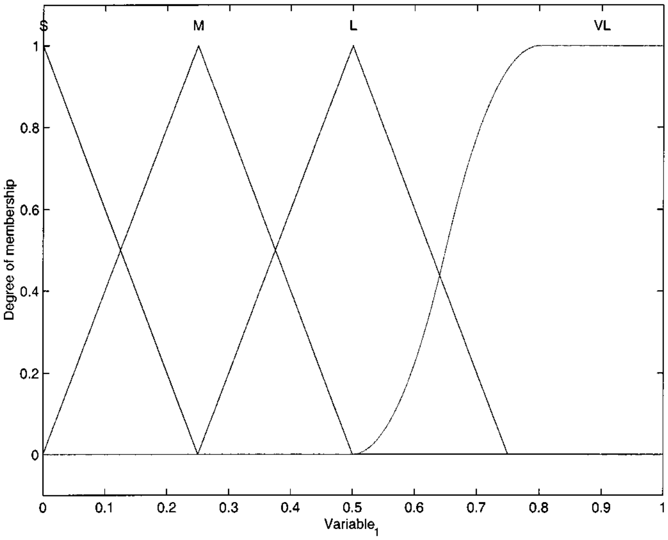


Fig. 3. Membership functions for the variables (normalized to fit in [0, 1]).

C. The Fuzzy Inference Systems

To implement the fuzzy algorithm for VCLQG weight selection, hereafter called the FV algorithm, two fuzzy inference systems are needed. The first fuzzy system implements the logic of Step 3b and the second fuzzy system implements the logic of Step 3c-i. These systems will be referred to, respectively, as the weight change (WC) FIS and the variance error change (VEC) FIS. The next sections describe these two fuzzy inference systems.

1) The Signs of the Fuzzy Variables: As already observed in the description of the FV algorithm, the WC FIS and VEC FIS are based on the magnitudes of the relevant variables. This results in fewer fuzzy rules and hence increases algorithm speed. Here we show how to determine the signs of the variables for which the FIS's output the magnitudes.

From the previous observations on the variation of the variance and its weight, it is obvious that the normalized sensitivity $\bar{\eta}_{ii}$ estimate is always negative (if properly computed) and that the polarity of the weight change $\Delta\alpha_i$ required to reduce ε_i is determined by the polarity of ε_i itself. Since a decrease in variance is associated with an increase in its weight and vice versa, for the WC FIS, a negative error ε_i input results in a negative weight change (decrease) $\Delta\alpha_i$ in the output and vice versa, i.e.,

$$\mathrm{sign}(\Delta\alpha_i) = \mathrm{sign}(\varepsilon_i) \qquad i = 1, 2, \ldots, q+m. \tag{61}$$

For $j \neq i$, the normalized sensitivity estimate $\bar{\eta}_{ji}$ can be either positive or negative. Therefore the output of the VEC FIS depends on the signs of both the weight change $\Delta\alpha_i$ and $\bar{\eta}_{ji}$. The sign relations among the inputs and outputs of the VEC FIS are summarized in Table I. As seen from Table I, the sign of the outputs for the VEC FIS is determined by the logical XOR operation on the signs of both inputs ($\Delta\alpha_i$ and $\bar{\eta}_{ji}$), i.e.,

$$\mathrm{sign}(\Delta\varepsilon_{j(i)}) = \mathrm{sign}(\Delta\alpha_i)\ \mathrm{XOR}\ \mathrm{sign}(\bar{\eta}_{ji}) \qquad i, j = 1, 2 \ldots, q+m. \tag{62}$$

2) Fuzzification of the Input and Output Variables: The variables (i.e., $|\bar{\eta}_{ij}|$, $|\varepsilon_*|$, and $|\Delta\alpha_*|$) are fuzzified into four membership groups and each group is given a linguistic label corresponding to its relative size in the universe of discourse (UD), i.e., the variable domain. Here, ε_* represents either ε_i, $\varepsilon_{j(i)}$, or $\Delta\varepsilon_{i(j(i))}$, and $\Delta\alpha_*$ represents either $\Delta\alpha_i$, $\Delta\alpha_{j(i)}$ or $\Delta\alpha_{i(j(i))}$. The UDs for $|\varepsilon_*|$ and $|\Delta\alpha_*|$ are determined, respectively, by (58) and (60).

Although a normalized sensitivity estimate $|\bar{\eta}_{ij}|$ is known to vary in the range $[0, \infty]$ corresponding to the possible absolute values of the gradient of the ε_i versus α_j curve, the fuzzy groups for $|\bar{\eta}_{ij}|$ are defined in the range [0, 1] where all values above 1 are absorbed in 1. The case for $|\bar{\eta}_{ij}| = 1$ occurs when the tangent to the ε_i versus α_j is inclined at ±45° with respect to the horizontal axis.

The UDs for the different variables are summarized in Table II. The resulting fuzzy groups together with their linguistic labels are shown in Table III, and membership functions which define the degree of membership for each of the fuzzy groups are shown in Fig. 3.

3) The Fuzzy Rule Sets: Both the WC FIS and VEC FIS are Mamdani-type [9], [19], [10] with two input variables and one output variable. Hence, the fuzzy rule sets are of the form

$$\textbf{IF}\ \text{(input variable 1)} \cdots \ \textbf{AND}\ \text{(input variable 2)} \cdots \ \textbf{THEN}\ \text{(output variable)}. \tag{63}$$

Since there are two sets of inputs, each with four membership groups, a total of 16 rules were developed for each FIS. These rules are given in the fuzzy associative matrices of Tables IV and V. These rules are developed based on experience.

TABLE IV
FUZZY ASSOCIATIVE MATRIX FOR THE WC FIS

$\lvert\varepsilon_i\rvert \setminus \lvert\eta_{ii}\rvert$	**S**	**M**	**L**	**VL**
S	S	S	S	S
M	L	M	M	S
L	VL	L	M	M
VL	VL	VL	L	L

TABLE V
FUZZY ASSOCIATIVE MATRIX FOR THE VEC FIS

$\lvert\Delta\alpha_i\rvert \setminus \lvert\eta_{ji}\rvert$	**S**	**M**	**L**	**VL**
S	S	S	S	S
M	S	M	M	L
L	M	M	L	VL
VL	L	L	VL	VL

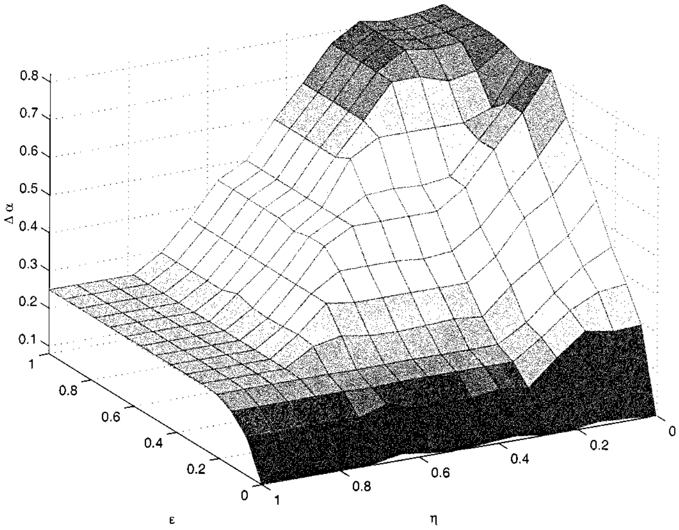


Fig. 4. The solution surface for the WC FIS.

4) Defuzzification of the Output Variable: For the outputs to have smooth transition between different output levels without any switching transients, it was implemented using the center of area method [9], [10], [19]. The corresponding solution surfaces for the two fuzzy systems are shown in Figs. 4 and 5. Fig. 4 is in agreement with the practice of a human designer in adjusting the weights and Fig. 5 is in agreement with the expected system behavior. As such, further tuning of the membership groups is not necessary.

D. Computational Issues

As noted in Step 3d of the fuzzy algorithm for VCLQG weight selection, the weight updates must not violate the definiteness constraints (5). $Q_x \geq 0$ is ensured after the update $\alpha_i \leftarrow \alpha_i + \Delta\alpha_i$ by making the following replacement for $i = 1, 2, \ldots q$:

$$\alpha_i \leftarrow \begin{cases} \alpha_i, & \text{if } \alpha_i > 0 \\ 0, & \text{otherwise.} \end{cases} \tag{64}$$

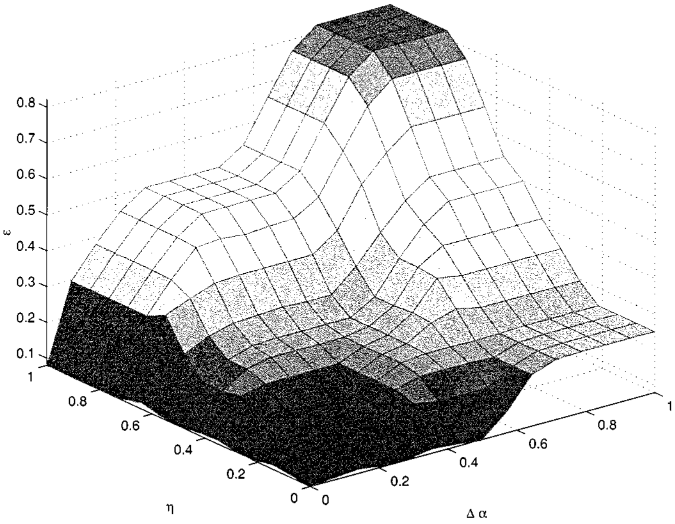


Fig. 5. The solution surface for the VEC FIS.

Similarly, $Q_u > 0$ is ensured by making the following replacements for $i = q+1, q+2, \ldots q+m$:

$$\alpha_i \leftarrow \begin{cases} \alpha_i, & \text{if } \alpha_i > 0 \\ \rho, & \text{otherwise} \end{cases} \tag{65}$$

where ρ is a small positive number. In the numerical experiments described in the next section, $\rho = 10^{-5}$ sufficed.

Recall that the sensitivities are estimated by changing the weights in small amounts $\delta\alpha_j$ and using the corresponding changes in variances δv_i to compute the sensitivity $\tilde{\eta}_{ij}$ using (54) and then the normalized sensitivity estimate $\overline{\eta}_{ij}$ using (55). The accuracy of $\overline{\eta}_{ij}$ is highly dependent on proper choice of $\delta\alpha_i$. In the algorithm implementation reported here, $\delta\alpha_i$ was chosen according to

$$\delta\alpha_i = \begin{cases} \alpha_i \times 10^{-6}, & \text{if } \alpha_i > 0 \\ 10^{-12}, & \text{otherwise.} \end{cases} \tag{66}$$

Three criteria were used to determine the algorithm convergence of Step 4 of the fuzzy algorithm for VCLQG weight selection. Let ε denote a vector of the variance errors, i.e.,

$$\varepsilon = [\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_{q+m}]^T. \tag{67}$$

The first convergence criterion is when ε is sufficiently small. In particular

$$e \triangleq \|\varepsilon\|_\infty \leq \epsilon_1 \tag{68}$$

where ϵ_1 is a small positive number.

It is quite possible that the constraint set is such that (68) cannot be satisfied. Hence, two additional criteria were developed to allow the algorithm to converge when the weight updates become ineffective. The first criterion is when the weights are very large in comparison to the updates, i.e., there exists a small positive number ϵ_2 such that

$$\left|\frac{\Delta\alpha_i}{\alpha_i}\right| \le \epsilon_2, \qquad i = 1, 2 \ldots, q + m. \tag{69}$$

The second criterion is when the weight updates cannot improve the retained error e, i.e., if Δe_k is the improvement in the retained error at each iteration k, defined as

$$\Delta e^{(k)} = e^{(k)} - e^{(k-1)} \tag{70}$$

then convergence is said to occur when

$$\left|\frac{\Delta e^{(k)}}{e^{(k)}}\right| \le \epsilon_3 \tag{71}$$

for some small positive number ϵ_3.

In summary the algorithm will converge in Step 4 if either (68), (69), or (71) are satisfied subject to (16) and (17). In all of the experiments conducted by the authors, the FV algorithm converged successfully by satisfying these conditions in less than 1 min. However, it is possible for the algorithm to fail in satisfying any of the above three convergence criteria. When this situation occurs, the algorithm will have to be stopped by using other means, such as a time limit or maximum number of iterations.

Numerical experiments are described in the next section. The convergence parameters for the FV algorithm were chosen as $\epsilon_1 = 5 \times 10^{-5}$, $\epsilon_2 = 1 \times 10^{-2}$ and $\epsilon_3 = 1 \times 10^{-1}$.

IV. NUMERICAL EXAMPLES AND PERFORMANCE RESULTS

The numerical experiments were performed using two systems. The first system is a stable two-input–two-output vibrating beam, adopted from [11], for which numerical experiments revealed that each LQG weight primarily influences the corresponding variance and has relatively small effect on the other variances. The second system is an unstable two-input–two-output hypothetical system for which each LQG weight tends to have a large influence on both the corresponding variance and some of the other variances. One set of experiments involved constraints on only the output variances, while a second set of experiments involved constraints on both the input and output variances.

Each of the algorithms was implemented in MATLAB and the fuzzy logic portion of the FV algorithm was implemented using the MATLAB Fuzzy Logic Toolbox [9]. The OCC algorithm was applied only to output constraints since it is only applicable to this case. The OVA algorithm requires both the input and output variances to be constrained and hence was applied only in this case. The NLP and FV algorithms were applied to all choices of the constraints.

When applying the OCC algorithm, the input weighting matrix, Q_u was chosen to be I_m. In order to use the NLP algorithm to solve the VCLQG problem, the weighting matrices defining the fixed quadratic cost function, Q_x and Q_u were chosen to be zero matrices.

TABLE VI
OPTIMAL PARAMETERS FOR THE NLP AND OCC ALGORITHMS USING CONSTRAINTS (74), (75)

NLP			OCC	
$\bar{\sigma}$	$\beta^{(0)}$	γ	$\bar{\gamma}$	β
0.9	0.5	10	5000	0.02

A. The System Models

The linear model (1)–(3) for vibrating model is given in [11] for the first two modes of vibration with the following system matrices:

$$A = \begin{bmatrix} 0.00 & 1.00 & 0.00 & 0.00 \\ -1.00 & -0.01 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 1.00 \\ 0.00 & 0.00 & -16.00 & -0.04 \end{bmatrix}$$

$$B = \begin{bmatrix} 0.0000 & 0.0000 \\ 0.5878 & -1.0000 \\ 0.0000 & 0.0000 \\ 0.9511 & 2.0000 \end{bmatrix}$$

$$C = E = \begin{bmatrix} 1.0000 & 0.00 & 2.0000 & 0.00 \\ 0.9511 & 0.00 & -0.5878 & 0.00 \end{bmatrix}$$

$$D_1 = 10^{-1.5} I_4, \qquad D_2 = 10^{-2} I_2. \tag{72}$$

For the hypothetical system, the following matrices were assumed:

$$A = \begin{bmatrix} -0.02 & 0.05 & 12.40 & -10.50 \\ -14.00 & 0.44 & -2.50 & -2.50 \\ -1.52 & 13.50 & -1.60 & 1.20 \\ 5.50 & -0.52 & 12.00 & 4.10 \end{bmatrix},$$

$$B = \begin{bmatrix} 1.00 & 3.50 \\ 0.59 & -1.50 \\ -5.20 & 2.00 \\ 0.95 & 1.55 \end{bmatrix}$$

$$C = E \begin{bmatrix} 1.00 & -0.50 & 2.50 & 0.35 \\ 0.95 & 2.50 & -0.59 & 1.40 \end{bmatrix}$$

$$D_1 = 10^{-3.5} I_4, \qquad D_2 = 10^{-2.5} I_2. \tag{73}$$

B. Optimization of the User Selected Parameters in the OCC and NLP Algorithms

The parameters $\bar{\gamma}$ and β in (46) of the OCC algorithm and $\bar{\sigma}$, $\beta^{(0)}$, and γ in (30) and (33) of the NLP algorithm must be chosen by the user. These parameters were chosen to optimize the performance of the respective algorithms for an output constrained variance control problem for the vibrating beam. In particular the output variances were constrained by

$$\lim_{t \to \infty} \mathbf{E}\left[z_1(t)^2\right] \le 0.0018 \tag{74}$$

$$\lim_{t \to \infty} \mathbf{E}\left[z_2(t)^2\right] \le 0.0020. \tag{75}$$

These variance constraints correspond to the output variances achieved when the output and input weights are, respectively

TABLE VII
SUMMARY OF THE PERFORMANCE OF THE FV, NLP, AND OCC ALGORITHMS FOR CONSTRAINTS ON INPUTS (RUNS = *Successful* RUNS, TIME IN SECONDS)

	$Q_z^{(0)} = I_2$				$Q_z^{(0)} = diag(10, 20)$				Overall
Alg.	Beam		Hypothetical		Beam		Hypothetical		Average
	Runs	Time/Run	Runs	Time/Run	Runs	Time/Run	Runs	Time/Run	Time/Run
FV	7	1.82	6	2.29	7	2.37	6	1.70	2.05
NLP	4	1.61	4	1.58	6	102.05	4	168.66	72.21
OCC	7	0.84	6	0.59	7	0.69	6	0.55	0.67

TABLE VIII
SUMMARY OF THE PERFORMANCE WITH VARIANCE CONSTRAINTS ON INPUTS AND OUTPUTS (RUNS = *Successful* RUNS, TIME IN SECONDS)

	Beam		Hypothetical		Average
Algorithm	Runs	Time/Run	Runs	Time/Run	Time/Run
FV	6	16.48	6	30.07	23.27
NLP	6	59.66	5	33.04	47.57
OVA	6	22.05	3	111.83	51.98

$Q_z = \mathrm{diag}(100, 0)$ and $Q_u = I_2$. The two output variance weights were each initialized to unity and the input variance weights were held fixed at unity. The NLP and OCC algorithms were used to determine output weights that would take the output variances to the boundary of the constraints. For both algorithms the space of user selected parameters was gridded and for each point in the grid the algorithm was allowed to converge. The grid for the parameters $\overline{\sigma}$ in the NLP algorithm and β in the OCC algorithm was created by varying each of the parameters from 1.5 to 5000 using the pattern: 1.5, 2, 3, 5, 10, $\ldots$. The grid for the parameters $\beta^{(0)}$ and γ in the NLP algorithm and $\overline{\gamma}$ in the OCC algorithm was generated by varying each parameter from 0.1 to 0.9 in increments of 0.1. For each point in the grid the time to convergence was recorded along with the obtained output variances. The parameters that yielded the lowest convergence times while bringing the output variances close to their boundaries are shown in Table VI. It took about 6 h to obtain the optimal OCC parameters and about 7 h to obtain the optimal NLP parameters. These parameters were held to these values throughout the numerical experimentation.

C. *Performance of the Algorithms*

Each algorithm was subjected to the convergence criteria (68), (69), and (71) and was forced to stop if at the end of a major iteration, the run time had exceeded 10 min. When the algorithm stopped due to either of the convergence criteria (68), (69), or (71) and the variance constraints (16) and (17) were satisfied, it was said to have successfully converged, and the convergence time (in seconds) was recorded. If the algorithm did converge by criteria (69) or (71) but did not satisfy (16) and (17), it was said to have been unsuccessfully converged and the results are not recorded. The algorithm was said to have not converged if it did not meet any of the convergence criteria and (16) and (17) were not satisfied. Once the NLP Algorithm failed due to numerical ill-conditioning and this was also treated as failure to converge.

Table VII summarizes the performance of each algorithm for the two systems when only the output variances were constrained and the output variance weights were set to two different initial conditions. It is seen from this table that the convergence time of the FV and OCC algorithms was essentially independent of the initial weights. However, the convergence time of the NLP algorithm varied dramatically with the initialization. It is possible that by reoptimizing the parameters $\overline{\sigma}$, $\beta^{(0)}$, and γ the performance of the NLP algorithm could be substantially improved for the $Q_z = \mathrm{diag}(10, 20)$ initialization. However, this is not practical since this optimization process is very time consuming. Table VIII compare the FV, NLP, and OVA algorithms for the two test systems when each of the output and input variances was constrained and all algorithms were initialized with unity weights. It was seen that whenever the algorithms converged, they yielded almost same weights. In all cases the FV and OCC algorithms always successfully converged. The NLP algorithm successfully converged 82% of the time while the OVA algorithm had a 76% success rate.

In general the FV algorithm performed favorably in comparison to the crisp algorithms. For the more difficult cases in which both the input and output variances were constrained, as shown in Table VIII, the average successful convergence times of the FV, NLP, and OVA algorithms were, respectively, 23.27 s, 47.57 s, and 51.98 s.

The dimensionality of the problem can be decreased by assigning a fixed value to one of the weights α_i. This option would help to reduce the number of solutions to the VCLQG problem and hence reduce the size of the solution set that needs to be searched. However, this can overly restrict the weight selection algorithms since they allow one or more of the weights to converge to zero. This can lead to numerical problems if the weight that is to be made zero is artificially restricted to a particular number. In addition, fixing a weight reduces the paths that a weight selection algorithm can take, which may also lead to numerical ill-conditioning. Hence, although the fixed-weight option should be explored in the future, it is not guaranteed to produce better results.

V. CONCLUSION

A fuzzy algorithm for automatic selection of the quadratic weights for VCLQG design has been presented. The algorithm converges substantially faster than the crisp algorithms which solve the general VCLQG problem. Its convergence time is also less dependent on the system, constraints, and the initial weights. This algorithm, however, is slower compared to the OCC algorithm when solving the VCLQG problem with only output variance constraints. Since for most practical problems, it is difficult to assign the variance constraints arbitrarily, a methodology to select these constraints reasonably is needed. It is recommended that the fuzzy methodologies developed here be extended further to be able accommodate the selection of reasonable variance constraints.

REFERENCES

[1] B. D. O. Anderson and J. B. Moore, *Optimal Control: Linear Quadratic Methods*. Engelwood Cliffs, NJ: Prentice-Hall, 1990.

[2] S. Boyd and C. Barratt, *Linear Controller Design: Limits of Performance*. Engelwood Cliffs, NJ: Prentice-Hall, 1991.

[3] A. E. Bryson, Jr. and Y.-C. Ho, *Applied Optimal Control: Optimization, Estimation and Control*. New York: Hemisphere, 1975.

[4] S. G. Cao, N. W. Rees, and G. Feng, "Analysis and design for a class of complex control systems, part II: Fuzzy controller design," *Automatica*, vol. 33, no. 6, pp. 1029–1039, 1997.

[5] C. W. Chan, K. C. Cheung, and C. X. Qin, "Fuzzy logic compensators for plants subject to actuator saturation," in *Proc. IFAC 1st Int. Workshop of Safety, Reliability, Applicat. Emerging Intell. Contr.*, Hong Kong, Dec. 1994, pp. 1–6.

[6] E. G. Collins, Jr. and M. F. Selekwa, "Fuzzy quadratic weights for variance constrained LQG design," in *Proc. 39rd Conf. Decision Contr.*, Phoenix, AZ, Dec. 1999, pp. 4044–4049.

[7] E. G. Collins, Jr. and R. E. Skelton, "A theory of state covariance assignment for discrete systems," *IEEE Trans. Automat. Contr.*, vol. 32, pp. 35–41, Jan. 1987.

[8] M. A. Franchek, "Selecting performance weights for the μ and H_∞ systhesis methods for SISO regulating systems," *Trans. ASME*, vol. 118, pp. 126–131, March 1996.

[9] N. Gulley and J. S. Jang, *Fuzzy Logic Toolbox User's Guide*. Natick, MA: The Mathworks Inc., 1995.

[10] C. J. Harris, C. G. Moore, and M. Brown, *Intelligent Control: Aspects of Fuzzy Logic and Neural Nets*. Singapore: World Scientific, 1993.

[11] C. Hsieh, R. E. Skelton, and F. M. Damra, "Minimum energy controllers with inequality constraints on output variances," *Opt. Contr. Apll. Meth.*, vol. 10, pp. 347–366, 1989.

[12] D. C. Hyland and D. S. Bernstein, "The optimal projection equations for fixed-order dynamic compensation," *IEEE Trans. Automat. Contr.*, vol. AC-29, pp. 1034–1037, Nov. 1984.

[13] P. P. Khargonekar and M. A. Rotea, "Multiple objective optimal control of linear systems: The quadratic norm case," *IEEE Trans. Automat. Contr.*, vol. 36, pp. 14–24, Jan. 1991.

[14] N. T. Koussoulas and C. T. Leondes, "The multiple linear quadratic Gaussian problem," *Int. J. Contr.*, vol. 43, no. 2, pp. 337–349, 1986.

[15] P. M. Mäkilä, "On multiple criteria stationary linear quadratic control," *IEEE Trans. Automat. Contr.*, vol. 34, pp. 1311–1313, Dec. 1989.

[16] P. M. Mäkilä, T. Westerlund, and H. T. Toivonen, "Constrained linear quadratic Gaussian control with process applications," *Automatica*, vol. 20, no. 4, pp. 15–29, 1984.

[17] G. Papageorgiou and K. Glover, "A systematic procedure for designing nondiagonal weights to facilitate $\mathcal{H}_\infty$ loop shaping," in *Proc. IEEE. 36th Conf. Decision Contr.*, San Diego, CA, Dec. 1997, pp. 2127–2132.

[18] K. M. Passino, "Intelligent control for autonomous systems," *IEEE Spectrum*, pp. 55–62, June 1995.

[19] K. M. Passino and S. Yurkovich, *Fuzzy Control*. Reading, MA: Addison Wesley Longman, 1998.

[20] R. E. Skelton and M. DeLorenzo, "Space structure control design by variance assignment," *J. Guidance*, vol. 8, no. 4, pp. 454–462, July 1985.

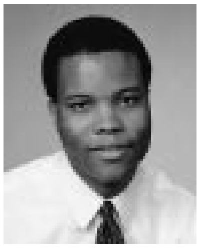
[21] H. T. Toivonen, "A multiobjective linear quadratic Gaussian control problem," *IEEE Trans. Automat. Cont.*, vol. AC-29, pp. 279–280, Mar. 1984.

[22] H. T. Toivonen and P. M. Mäkilä, "Computer aided design procedure for multiobjective LQG control problems," *Int. J. Control*, vol. 49, no. 2, pp. 655–666, 1989.

[23] L. X. Wang, "Stable and optimal fuzzy control of linear systems," *IEEE Trans. Fuzzy Syst.*, vol. 16, pp. 137–143, Feb. 1998.

[24] H. Ying, "Practical design of nonlinear fuzzy controllers with stability analysis for regulating processes with unknown mathematical model," *Automatica*, vol. 30, no. 7, pp. 1185–1195, 1994.

[25] G. Zhu, M. A. Rotea, and R. E. Skelton, "A convergent algorithm for the output covariance constraint control problem," *SIAM J. Cont. Optim.*, vol. 35, no. 1, pp. 341–361, January 1997.



Emmanuel G. Collins, Jr. (S'83–M'86–SM'99) received the B.S. degree from Morehouse College, Atlanta, GA, in 1981, the B.M.E. degree from the Georgia Institute of Technology, Atlanta, in 1981, the M.S. degree in mechanical engineering from Purdue University, West Lafayette, IN, in 1982, and the Ph.D. degree in aeronautics and astronautics from Purdue University in 1987.

He worked for seven years in the Controls Technology Group at Harris Corporation, Melbourne, FL, where he specialized in the development and application of modern robust control theories to the control of flexible spacecraft. He joined the Mechanical Engineering Department at Florida A&M University, Florida State University College of Engineering, Tallahassee, where he currently serves as Professor. His recent work has focused on control of manufacturing processes, automated controller tuning, fault detection and isolation, automated weight selection in modern control, numerical algorithms for control law design, fluidic thrust vector control, and intelligent control.

Majura F. Selekwa received the B.Sc.(Eng.) degree in mechanical engineering from the University of Dar es Salaam, Tanzania, and the M.Eng.Sc degree in mechatronics from the University of New South Wales, Australia. He received the Ph.D. degree in 2001 from Florida A&M University, Tallahassee.

He had been a faculty in the Department of Mechanical Engineering at the University of Dar es Salaam from 1989 to fall 1997, when he became a Research/Teaching Assistant in Mechanical Engineering Department at the FAMU-FSU College of Engineering while working toward the Ph.D. degree. Currently he is a Postdoctoral Research Associate in Mechanical Engineering at the FAMU-FSU College of Engineering. His research interests include the application of intelligent methods in the design of multivariable controllers, mechatronic systems, numerical optimization, industrial robotics, and nonlinear and robust control systems.