

Feedback Control Methodologies for Nonlinear Systems¹

S. C. Beeler², H. T. Tran³, and H. T. Banks⁴

Center for Research in Scientific Computation
Department of Mathematics
North Carolina State University
Raleigh, NC 27695

Abstract. A number of computational methods have been proposed in the literature to design and synthesize feedback controls when the plant is modeled by nonlinear dynamics. It is not immediately clear, however, which is the best method for a given problem; this may depend on the nature of the nonlinearities, the size of the system, whether the amount of control used or time needed for the method is a concern, and other factors. In this paper a comprehensive comparison study of five methods for the synthesis of nonlinear control systems is carried out. The performance of the methods on several test problems are studied, and some recommendations are made as to which feedback control method is best to use under various conditions.

¹This research was supported in part by a DOD/AFOSR MURI Grant AFOSR F49620-95-1-0447 and in part by AFOSR Grant F49620-96-1-0292 (AASERT)

²e-mail: scbeeler@eos.ncsu.edu

³e-mail: tran@control.math.ncsu.edu

⁴e-mail: htbanks@eos.ncsu.edu

1. Introduction

The optimal feedback control of a linear system is a subject which has been extensively studied (see e.g. [1] and the references therein). If the cost functional is quadratic in the state and control, and we assume full state knowledge, then the optimal control is a linear state feedback law where the control gains are obtained by solving a differential/algebraic Riccati equation. The success of this linear quadratic regulator (LQR) problem is due to the successful development of robust and efficient algorithms for solving the Riccati equation. However, if the plant is described by nonlinear dynamics, then the optimal state feedback law is given in terms of the solution to the Hamilton-Jacobi-Bellman (HJB) equation [2]. The HJB equation provides the solution to the optimal control problem for general nonlinear systems; however, it is in most cases impossible to solve analytically. This has led to many methods being proposed in the literature for ways to approximately obtain the solution to the HJB equation as well as obtain a suboptimal feedback control for general nonlinear dynamical models.

One possibility is to construct the control as a power series, either by separating out the nonlinearities in the system into a power series, or by introducing a temporary variable and expanding around it. Then the first few terms in the series for the control are found by various techniques. This idea is based on considering the system as a perturbation of a linear system, with the control being an extension of the linear control (the first term of the power series is obtained by solving the Riccati equation for the solution of the linearized system). These types of methods are described in [3, 4, 5, 6].

Another approach is through successive approximation, where an iterative process is used to find a sequence of approximations approaching the solution of the HJB equation. This is done by solving a sequence of generalized Hamilton-Jacobi-Bellman (GHJB) equations, and is discussed in [7, 8] in a general context. A more concrete technique for finding the desired solution is described in [9, 10], where a Galerkin procedure is used to find a numerical solution to the GHJB equation.

Other methods include the state-dependent Riccati equation (SDRE), which is an extension of the Riccati equation to nonlinear systems [11, 12, 13]. The coefficients in the SDRE are functions of the state instead of being constant-valued as in the linear case. This makes the equation much more difficult to solve. Also, in the nonlinear case the resulting control is only suboptimal. There is also a control method based on representing the curve of the system's solution as a linear operator and then inverting it [14], and an exploration of the use of neural nets to solve the HJB equation [15].

Open-loop controls can also be found by a variety of different methods. While open-loop controls are generally not useful for practical applications, it is possible to extend them into closed-loop controls. This can be done with an interpolation over the state space as described in [16], for example. The open-loop control can be found by solving a two-point boundary value (TPBV) problem, which is obtained from the necessary optimality conditions. It can be solved by discretizing and then solving exactly [16], or else by one of several iterative methods (quasilinearization, gradient methods, etc.) as in [17, 18, 19, 20, 21].

It is also important to note that different methods are applicable to different problems. Some of the methods that we will investigate in detail can be used on a more general problem than the one we will consider, for example, with a nonlinear function of the control as well as the state. There are also methods which are designed for other specific problems, such as the bilinear system $\dot{x} = Ax + Bu + (\sum x_j N_j)u$. Methods to solve this type of problem are discussed in [22, 23, 24, 25, 26]. Another

variation on the nonlinear control problem is an uncertain system, where unknown parameters are involved in the equation, as discussed in [27, 28]. A method dealing only with systems where the number of state and control variables are the same is presented in [29].

In this paper a comprehensive comparison study of five of the methods described above will be performed on several test problems. We will draw some conclusions as to which methods give the best results for certain types of problems, and also consider other factors such as the flexibility of the methods and the difficulty and time involved in their implementation and use. The paper is organized as follows. The problem statement is given in Section 2. The methods to be considered will be described in Section 3. In particular, Section 3.3 describes the successive approximation method. Because there was an error in one of the formulas in the original paper, this section contains a detailed derivation of the method. The performance of the methods on four examples are discussed in Section 4, and overall conclusions are given in Section 5.

2. Problem Statement

Consider a nonlinear system of the type

$$\begin{cases} \dot{x}(t) &= f(x(t)) + Bu(x(t)) \\ x(0) &= x_0, \end{cases} \quad (1)$$

where $f(x)$ is a nonlinear function of the state $x \in \Omega \subset \mathbb{R}^m$, $u : \Omega \rightarrow \mathbb{R}^k$ is the control, and B is a constant-valued $m \times k$ matrix. Also consider a cost functional

$$J(x_0, u) = \int_0^\infty (x^T Q x + u^T R u) dt$$

with a given constant-valued $m \times m$ symmetric positive semi-definite matrix Q and a $k \times k$ symmetric positive definite matrix R . The optimal control problem is to find a state feedback control $u^*(x)$ which minimizes the cost for all possible initial conditions x_0 .

For the simpler linear problem, where $f(x) = A_0 x$, the optimal feedback control is given by $u^*(x) = -R^{-1} B^T P x$, with P being the solution to the matrix Riccati equation $PA_0 + A_0^T P - PBR^{-1}B^T P + Q = 0$. The theories for this linear quadratic regulator (LQR) problem have been established for both the finite and infinite dimensional problems (see e.g. [30, 31]). In addition, stable and robust algorithms for solving the Riccati equation have already been developed and are well documented in many places in the literature and in textbooks.

For the nonlinear case, the optimal feedback control is known to be of the form

$$u^*(x) = -\frac{1}{2} R^{-1} B^T V_x(x),$$

where the function V is the solution to the Hamilton-Jacobi-Bellman equation

$$V_x^T(x)f(x) - \frac{1}{4} V_x^T(x) B R^{-1} B^T V_x(x) + x^T Q x = 0. \quad (2)$$

The HJB equation itself is very difficult to solve analytically for any but the simplest problems, however. Thus efforts have been made to numerically approximate the solution of the HJB equation, or to solve a related problem producing a suboptimal control, or to use some other process in order

to obtain a usable feedback control. The following section will outline some such methods that have been proposed in the literature.

3. Feedback Control Methodologies for Nonlinear Systems

3.1. Power Series Approximation

One type of method of obtaining an approximate solution to the HJB equation (2) is by using a power series to approximate $V(x)$. This is done by Garrard and others in [3, 4, 5]. Consider the representation $V(x) = \sum_{n=0}^{\infty} V_n(x)$, where each $V_n(x) = O(x^{n+2})$. Also separate the nonlinear function $f(x)$ into powers of x as $f(x) = A_0x + \sum_{n=2}^{\infty} f_n(x)$ with $f_n(x) = O(x^n)$. Substituting these expansions into the HJB equation results in

$$\left(\sum_{n=0}^{\infty} (V_n)_x^T \right) \left(A_0x + \sum_{n=2}^{\infty} f_n(x) \right) - \frac{1}{4} \left(\sum_{n=0}^{\infty} (V_n)_x^T \right) B R^{-1} B^T \left(\sum_{n=0}^{\infty} (V_n)_x \right) + x^T Q x = 0.$$

By separating out by powers of x we obtain a series of equations:

$$(V_0)_x^T A_0 x - \frac{1}{4} (V_0)_x^T B R^{-1} B^T (V_0)_x + x^T Q x = 0 \quad (3)$$

$$(V_1)_x^T A_0 x - \frac{1}{4} (V_1)_x^T B R^{-1} B^T (V_0)_x - \frac{1}{4} (V_0)_x^T B R^{-1} B^T (V_1)_x + (V_0)_x^T f_2(x) = 0 \quad (4)$$

$$(V_n)_x^T A_0 x - \frac{1}{4} \sum_{k=0}^n [(V_k)_x^T B R^{-1} B^T (V_{n-k})_x] + \sum_{k=0}^{n-1} [(V_k)_x^T f_{n+1-k}(x)] = 0, \quad (5)$$

where $n = 2, 3, 4, \dots$

Equation (3) can be solved with $V_0(x) = x^T P x$, where the symmetric positive definite matrix P solves the Riccati equation $P A_0 + A_0^T P - P B R^{-1} B^T P + Q = 0$. This gives the standard linear control. Equations (4) and (5) can be solved for V_n , $n = 1, 2, 3, \dots$, by making V_n a scalar polynomial containing all possible combinations of products of the state elements with a total order of $n+2$, and then substituting it into the equation and solving for the coefficients. This can be done for as many terms as desired, but in general this quickly gets very complicated, especially for problems with a large number of state variables. As a way of avoiding this, Garrard proposes a method of finding $(V_1)_x$ more simply, which results in a very easy method of finding a quadratic type control [3].

Instead of using the polynomial representation, we may consider (4) as is and use the substitution $(V_0)_x = 2Px$ (where P is the solution to the Riccati equation) to obtain

$$(V_1)_x^T A_0 x - \frac{1}{4} (V_1)_x^T B R^{-1} B^T (2Px) - \frac{1}{4} (2x^T P) B R^{-1} B^T (V_1)_x + (2x^T P) f_2(x) = 0.$$

Rearranging some terms, we find

$$x^T \left[A_0^T (V_1)_x - P B R^{-1} B^T (V_1)_x + 2P f_2(x) \right] = 0.$$

The quantity inside the brackets is zero when $(V_1)_x = -2(A_0^T - P B R^{-1} B^T)^{-1} P f_2(x)$. This along with the $(V_0)_x$ term gives a quadratic feedback control law of the form

$$u(x) = -R^{-1} B^T [Px - (A_0^T - PBR^{-1} B^T)^{-1} P f_2(x)]. \quad (6)$$

If $f_2(x) = 0$ (that is, if there are no quadratic terms), then (4) is solved trivially by $V_1 = 0$. Equation (5) for $n = 2$ will then be of the form

$$(V_2)_x^T A_0 x - \frac{1}{4} (V_2)_x^T B R^{-1} B^T (V_0)_x - \frac{1}{4} (V_0)_x^T B R^{-1} B^T (V_2)_x + (V_0)_x^T f_3(x) = 0.$$

This is exactly the same form as (4) except that now V_2 is in the place of V_1 and f_3 is in the place of f_2 . Thus the solution is analogous to that of (4), with $(V_2)_x = -2(A_0^T - PBR^{-1} B^T)^{-1} P f_3(x)$, resulting in a feedback control of the form

$$u(x) = -R^{-1} B^T [Px - (A_0^T - PBR^{-1} B^T)^{-1} P f_3(x)].$$

If $f_3 = 0$ as well, then the resulting control will be the same except for f_4 in the place of f_3 . In other words, the two-term control described by Garrard is found using the lowest-order nonzero f_n term, whatever that may be.

This method has some limitations. It is not possible to increase the accuracy of the approximation by adding more terms to the power series without the tedious process of solving for the polynomial of the appropriate order. Only the first two terms of the series can be found by matrix calculations as described above. Also, the two-term control only considers one order of nonlinearity, as shown in (6) where only f_2 is used, ignoring any higher order parts of f . If a problem has both quadratic and cubic nonlinearities, for example, then the cubic part of the system will be ignored. Because of this property it seems reasonable to only use this method on problems with one order of nonlinearity (whether this order is quadratic, cubic, or higher). The method does have the advantage that it is very easy to implement, and its calculations can be done very quickly.

3.2. State-Dependent Riccati Equation

This method also results in a feedback control in power series form, though it does so through a different process. Instead of solving the HJB equation itself, the power series expansion is applied to the related state-dependent Riccati equation (SDRE). The SDRE method is discussed in detail by Cloutier, D'Souza and Mracek in [13], and the use of the power series for this equation is from Wernli and Cook [11].

The idea behind the method is to parallel the use of the Riccati equation for linear problems by rewriting the nonlinear function of x in (1) as $f(x) = A(x)x$. Note that the choice of A is not unique, and different choices will result in different controls. With f rewritten in this way the state-dependent Riccati equation is of the form

$$P(x)A(x) + A^T(x)P(x) - P(x)BR^{-1}B^T P(x) + Q = 0, \quad (7)$$

and the optimal feedback control is given by

$$u(x) = -R^{-1} B^T P(x)x. \quad (8)$$

We note that the Riccati solution $P(x)$ is state-dependent and is not as easy to find as for the constant coefficient case, except for simple problems with certain structures. One proposed method for the general case is to use the power series expansion to approximate the solution.

We rewrite A in terms of a constant part and a state-dependent part as $A(x) = A_0 + \varepsilon \Delta A(x)$ where ε is a temporary variable used for the expansion that will be set to 1 later. We next write P as a power series in ε :

$$\begin{aligned} P(x, \varepsilon) &= P(x)|_{\varepsilon=0} + P_\varepsilon(x)|_{\varepsilon=0}\varepsilon + P_{\varepsilon\varepsilon}(x)|_{\varepsilon=0}\frac{\varepsilon^2}{2} + \dots \\ &= \sum_{n=0}^{\infty} \varepsilon^n L_n(x), \end{aligned} \quad (9)$$

where P is symmetric, and so each L_n is symmetric as well. Substituting these expansions into the state-dependent Riccati equation (7), we find

$$\begin{aligned} \left(\sum_{n=0}^{\infty} \varepsilon^n L_n \right) (A_0 + \varepsilon \Delta A(x)) + \left(A_0^T + \varepsilon \Delta A^T(x) \right) \left(\sum_{n=0}^{\infty} \varepsilon^n L_n \right) \\ - \left(\sum_{n=0}^{\infty} \varepsilon^n L_n \right) B R^{-1} B^T \left(\sum_{n=0}^{\infty} \varepsilon^n L_n \right) + Q = 0. \end{aligned}$$

Now by matching terms involving the same powers of ε we obtain the following set of equations that can be used to determine L_n

$$L_0 A_0 + A_0^T L_0 - L_0 B R^{-1} B^T L_0 + Q = 0 \quad (10)$$

$$L_1 \left(A_0 - B R^{-1} B^T L_0 \right) + \left(A_0^T - L_0 B R^{-1} B^T \right) L_1 + L_0 \Delta A + \Delta A^T L_0 = 0 \quad (11)$$

$$\begin{aligned} L_n \left(A_0 - B R^{-1} B^T L_0 \right) + \left(A_0^T - L_0 B R^{-1} B^T \right) L_n + L_{n-1} \Delta A + \Delta A^T L_{n-1} \\ - \sum_{k=1}^{n-1} \left(L_k B R^{-1} B^T L_{n-k} \right) = 0. \end{aligned} \quad (12)$$

Equation (10) is the standard Riccati equation for the linear part A_0 . Equations (11) and (12) may be solved, but as with the HJB power series expansion it may be a tedious process if the function $\Delta A(x)$ is complicated. However, in some special cases, it is easier to obtain higher-order results for the SDRE expansion. For example, in the case that the nonlinear function $f(x)$ in (1) is quadratic in x , the function $\Delta A(x)$ will be linear in x , and from (11) the L_1 term will be of order x which will produce an order x^2 control, and so on. In comparison to the HJB expansion, it requires the order x^3 term V_1 to obtain an order x^2 control; thus it is necessary to consider a much larger number of possible combinations of the state variables in x .

The SDRE method is much easier to consider for a certain class of nonlinearity factorizations, specifically, those for which ΔA has the same function of x in all of its elements, and so can be written as $\Delta A(x) = g(x) \Delta A_C$ with a constant-valued matrix ΔA_C . In this case, by defining $L_n(x) = g^n(x) (L_n)_C$, where $(L_n)_C$ is a constant matrix, we obtain from (12)

$$\begin{aligned} (L_n)_C \left(A_0 - B R^{-1} B^T L_0 \right) + \left(A_0^T - L_0 B R^{-1} B^T \right) (L_n)_C + (L_{n-1})_C \Delta A_C + \Delta A_C^T (L_{n-1})_C \\ - \sum_{k=1}^{n-1} \left((L_k)_C B R^{-1} B^T (L_{n-k})_C \right) = 0. \end{aligned}$$

This is just a constant-valued matrix equation, which can be solved easily. In this way a large class of SDRE problems can be solved to as many terms of the power series as desired. Once as many L_n terms as desired have found, the control is obtained by substituting them back into (9) and (8) with ε set to 1.

The SDRE method is similar in its advantages and disadvantages to the two-term HJB method. They both consist of calculating out the first few terms of a power series, and can both be calculated very quickly. However, as many terms of the SDRE series as desired can be found for the case when $\Delta A = g(x)\Delta A_C$, as described above, while the HJB approach always uses only two terms. This assumption on the form of ΔA does limit the problems for which the SDRE approach is most useful, just as the HJB approach is most useful only for problems with only one level of nonlinearity. There is also the drawback that the SDRE method is approximating the solution of the SDRE, which would produce only a suboptimal control even if solved exactly, while the HJB method is approximating the HJB equation itself.

3.3. Successive Galerkin Approximation

Instead of the regular HJB equation (2), the successive Galerkin approximation (SGA) method (from Beard, Saridis and Wen in [9, 10]) uses the generalized Hamilton-Jacobi-Bellman (GHJB) equation. Because there is an error in the paper by Beard et al. [10] on page 602, we rederive the proposed method here. Consider the cost functional

$$V(x_0; u) = J(x_0, u) = \int_0^\infty (x^T Q x + u^T R u) dt$$

where $x(t)$ satisfies the nonlinear dynamics $\dot{x} = f(x) + Bu(x)$, with control $u(x)$ and initial condition x_0 . Differentiating V along the path of the system leads to the GHJB equation,

$$GHJB[V, u] \equiv \left(\frac{\partial V}{\partial x_0} \right)^T (f(x_0) + Bu(x_0)) + x_0^T Q x_0 + u^T(x_0) R u(x_0) = 0. \quad (13)$$

This becomes the normal HJB equation if the formula for the control,

$$u(x_0) = -\frac{1}{2} R^{-1} B^T \frac{\partial V}{\partial x_0}(x_0), \quad (14)$$

is used in (13). Instead, a process is set up to solve (13) and (14) together in an iterative process.

We first choose an initial control function $u^{(0)}(x)$ (for example, the linear optimal control). Then we use the GHJB equation (with the variable x_0 now changed to x) to find $\frac{\partial V^{(0)}}{\partial x}$ by solving

$$\left(\frac{\partial V^{(0)}}{\partial x} \right)^T (f(x) + Bu^{(0)}(x)) + x^T Q x + (u^{(0)})^T R u^{(0)} = 0.$$

We let the next control in the sequence be

$$u^{(1)}(x) = -\frac{1}{2} R^{-1} B^T \frac{\partial V^{(0)}}{\partial x}(x).$$

We proceed to use this to find $V^{(1)}$ and $u^{(2)}$, and continue in this way for as long as desired.

A Galerkin method can be used to approximate the GHJB equation so that it can be solved numerically. To do this, we choose a set of N basis functions $\{\phi_j\}_{j=1}^N$ and define the L_2 inner product as

$$\langle f, g \rangle = \int_{\Omega} f(x)g(x)dx,$$

where Ω is any compact subset of the region of attraction associated with the known stabilizing control $u^{(0)}(x)$. The Galerkin approximation of $V^{(i)}$ is given by

$$V_N^{(i)}(x) = \sum_{j=1}^N c_j^{(i)} \phi_j(x),$$

where the coefficients $c_j^{(i)}$ satisfy the weak form of (13):

$$\langle GHJB[\sum_{j=1}^N c_j^{(i)} \phi_j, u_N^{(i)}], \phi_n \rangle = 0 \quad (15)$$

for $n = 1, \dots, N$. The initial control function $u_N^{(0)}$ for (15) is chosen, and the later iterates are found as follows. Equation (15) gives N equations to solve for the N unknowns $c_j^{(i)}$, $j = 1, 2, \dots, N$:

$$\sum_{j=1}^N c_j^{(i)} \left\langle \left(\frac{\partial \phi_j}{\partial x} \right)^T (f + Bu_N^{(i)}), \phi_n \right\rangle + \langle x^T Qx + (u_N^{(i)})^T Ru_N^{(i)}, \phi_n \rangle = 0 \quad (16)$$

for $n = 1, \dots, N$. The next control update becomes

$$u_N^{(i+1)} = -\frac{1}{2}R^{-1}B^T \sum_{j=1}^N c_j^{(i)} \left(\frac{\partial \phi_j}{\partial x} \right). \quad (17)$$

From (16) and (17) the process can be written as an iterative matrix problem, with the necessary inner products calculated beforehand. We will use the notation $C_N^{(i)} = (c_1^{(i)}, \dots, c_N^{(i)})^T$, $\Phi_N(x) = (\phi_1(x), \dots, \phi_N(x))^T$, and let $\nabla \Phi_N$ denote the Jacobian of Φ_N . Also, for any real-valued function $\eta(x)$, define a vector of inner products

$$\langle \eta, \Phi_N \rangle_v = (\langle \eta, \phi_1 \rangle, \dots, \langle \eta, \phi_N \rangle)^T,$$

and for any vector-valued function $\eta(x) = (\eta_1(x), \dots, \eta_N(x))^T$, define a matrix of inner products

$$\langle \eta, \Phi_N \rangle_m = \begin{bmatrix} \langle \eta_1, \phi_1 \rangle & \cdots & \langle \eta_N, \phi_1 \rangle \\ \vdots & \ddots & \vdots \\ \langle \eta_1, \phi_N \rangle & \cdots & \langle \eta_N, \phi_N \rangle \end{bmatrix}.$$

Compute the following L_2 inner products:

$$I^Q = \langle x^T Qx, \Phi_N \rangle_v, I^R = \langle (u^{(0)})^T Ru^{(0)}, \Phi_N \rangle_v, I^f = \langle \nabla \Phi_N f, \Phi_N \rangle_m, \quad (18)$$

$$I^B = \langle \nabla \Phi_N Bu^{(0)}, \Phi_N \rangle_m, \{I_k^{BRB}\}_{k=1}^N = \left\{ \langle \nabla \Phi_N BR^{-1}B^T (\partial \phi_k / \partial x), \Phi_N \rangle_m \right\}_{k=1}^N. \quad (19)$$

From the chosen initial iterate $u^{(0)}$ solve

$$Ac_N^{(0)} = b,$$

where $A = I^f + I^B$ and $b = -I^Q - I^R$. Given the (i) th iterate, the $(i + 1)$ th iterate is the solution to the equation

$$Ac_N^{(i+1)} = b,$$

where $A = I^f - \frac{1}{2} \sum_{k=1}^N c_k^{(i)} I_k^{BRB}$ and $b = -I^Q - \frac{1}{4} \left(\sum_{k=1}^N c_k^{(i)} I_k^{BRB} \right) C_N^{(i)}$. (Note the $\frac{1}{4}$ in the expression for b , which was incorrectly written as $\frac{1}{2}$ in the original paper [10].) At each step $i \geq 0$, the control is given by

$$u_N^{(i+1)} = -\frac{1}{2} R^{-1} B^T \nabla \Phi_N^T C_N^{(i)}.$$

The successive Galerkin approximation method is more time-intensive than the first two methods discussed. The actual iteration process is fast, but the time needed to calculate the integrals in (18) and (19) can be significant. The method has the advantage that it is applicable to a larger class of problems, without the restrictions imposed by the first two methods. However, it has the disadvantage, being an iterative method, that it is dependent on the chosen initial iterate $u^{(0)}$. If this is not well chosen, then the method may converge very slowly, or may even not converge at all.

3.4. Interpolation of TPBV Problem Solutions

A very different approach to finding a feedback control is used in the remaining two methods to be discussed next. These methods use techniques for finding an open-loop control for the system with a chosen initial condition x_0 . This is repeated for many values of x_0 . At each point, the initial control $u^{(x_0)}(t = 0)$ that is found is considered the control value at x_0 . Then an interpolation is done over all these initial points to form a closed-loop (feedback) control $u(x)$. The open-loop control is found by solving the Hamiltonian system arising from the necessary optimality conditions [1, 17] given by

$$\begin{cases} \dot{x} &= f(x) - \frac{1}{2} B R^{-1} B^T p \\ \dot{p} &= -f_x^T(x) p - 2Qx \end{cases} \quad (20)$$

with initial condition $x(0) = x_0$ and final condition $\lim_{t \rightarrow \infty} p(t) = 0$. From this, the open-loop control is calculated as

$$u(t) = -\frac{1}{2} R^{-1} B^T p(t).$$

In the version of this method described by Ito and Schroeter [16], this problem is solved as follows. First the final time is changed from infinity to a chosen finite T , then the two-point boundary value (TPBV) problem is discretized numerically for $t \in [0, T]$. The discrete TPBV problem is given by

$$\begin{cases} \frac{1}{\Delta t} (x_k - x_{k-1}) &= \frac{1}{2} (f(x_k) + f(x_{k-1})) - \frac{1}{2} B R^{-1} B^T p_k \\ \frac{1}{\Delta t} (p_{k+1} - p_k) &= -\frac{1}{2} f_x^T(x_k) (p_k + p_{k+1}) - 2Qx_k \end{cases} \quad (21)$$

where $k = 1, \dots, N - 1$, and N is a prescribed number of discretization intervals, so that $\Delta t = T/N$. Here x_0 is the chosen initial condition, and $p_N = 0$.

Letting $y = (x_1, \dots, x_{N-1}, p_1, \dots, p_{N-1})$ and rewriting (21) as

$$F(y) = \left(\begin{array}{c} \frac{1}{\Delta t}(x_k - x_{k-1}) - \frac{1}{2}(f(x_k) + f(x_{k-1})) + \frac{1}{2}BR^{-1}B^T p_k \\ \frac{1}{\Delta t}(p_{k+1} - p_k) + \frac{1}{2}f_x^T(x_k)(p_k + p_{k+1}) + 2Qx_k \end{array} \right)_{k=1, \dots, N-1} = 0 \quad (22)$$

results in a $2m(N - 1)$ dimensional system of nonlinear equations (where m is the number of state variables) which can be solved using Newton's method. The Newton's method iterate is

$$y_+ = y_c - [\nabla F(y_c)]^T F(y_c),$$

where ∇F is the Jacobian of F and has the following structure:

$$\nabla F = \left(\begin{array}{cc} A & C \\ D & -A^T \end{array} \right).$$

The matrix A is block lower bi-diagonal and the matrices C and D are block diagonal, all with block size m . The values of the blocks are as follows (using I_m as an $m \times m$ identity matrix):

$$\begin{aligned} A_{k,k} &= \frac{1}{\Delta t}I_m - \frac{1}{2}f_x(x_k), \\ A_{k+1,k} &= -\frac{1}{\Delta t}I_m - \frac{1}{2}f_x(x_k), \\ C_{k,k} &= \frac{1}{2}BR^{-1}B^T, \\ D_{k,k} &= \frac{1}{2}f_{xx}^T(x_k)(p_k + p_{k+1}) + 2Q. \end{aligned}$$

We have carried out the required interpolations, following Ito and Schroeter's choice, with Green's functions of type $G(x, z) = |x - z|^\alpha$ and a choice of $\alpha = 3.7$ for the exponent. The closed-loop control is then given by

$$u(x) = \sum_{i=1}^M \eta_i G(x, (x_0)_i),$$

where $\{\eta_i\}$ are the coefficients of the interpolation and $\{(x_0)_i\}$ are the M initial points used to interpolate over the region Ω . The coefficients are chosen so that the control fits exactly at the interpolation points, so that

$$u((x_0)_j) = \sum_{i=1}^M \eta_i G((x_0)_j, (x_0)_i) = u_{OL}(t = 0, (x_0)_j)$$

for all $j = 1, \dots, M$. Here $u_{OL}(t = 0, (x_0)_j)$ is the open-loop control for initial condition $(x_0)_j$, evaluated at $t = 0$. The coefficients $\{\eta_i\}$ that will achieve this exact fit at the interpolation points are found by solving the $2M$ dimensional linear system

$$\begin{bmatrix} I_K G((x_0)_1, (x_0)_1) & \cdots & I_K G((x_0)_1, (x_0)_M) \\ \vdots & \ddots & \vdots \\ I_K G((x_0)_M, (x_0)_1) & \cdots & I_K G((x_0)_M, (x_0)_M) \end{bmatrix} \begin{bmatrix} \eta_1 \\ \vdots \\ \eta_M \end{bmatrix} = \begin{bmatrix} u_{OL}(t=0, (x_0)_1) \\ \vdots \\ u_{OL}(t=0, (x_0)_M) \end{bmatrix},$$

where I_K is an identity matrix of dimension K , the number of control variables. Note that the values on the diagonal are all zero for the chosen function G .

The use of the interpolation makes this method very time-consuming, given that the open-loop control must be found for many different initial states. This is especially true for problems with larger numbers of state variables; this increases the time needed both by making the discretized TPBV problem much larger and by increasing the number of interpolation points needed due to the larger dimension. It is not a very complicated process, however, and all of the expensive computation is done off-line, so the actual control of the system can still be done very quickly. There are no restrictions on the types of problem that can be solved. The process of increasing the accuracy of the control by adding more interpolation points is also straightforward.

3.5. Interpolation of Iterative Solutions

As in the previous method, this technique also uses interpolation over a number of open-loop control solutions, but the method of obtaining these solutions is different. In this method, proposed by Markman and Katz [20, 21], the Hamiltonian system is solved not as one large discretized TPBV problem given by (22), but through an iteration process leading to the desired control.

We again choose a final time T , and with it define a distance function

$$F(T, p_0) = \|x(T; x_0, p_0)\|^2 + \|p(T; x_0, p_0)\|^2 = \|z(T; z_0)\|^2.$$

Here $z = (x, p)^T$ is used to simplify notation, and $z(T; z_0)$ is the solution at time T of the Hamiltonian system (20) starting at z_0 (and $x(T; x_0, p_0)$ and $p(T; x_0, p_0)$ are the state and costate parts of this solution). F is a measurement of how close to zero the system will be at time T , given a value of p_0 . Thus the open-loop control is found by solving $\inf_{p_0} F(T, p_0)$ to find the value of p_0 which minimizes F (and so drives the system closest to zero at time T). The minimization is done by a Gauss-Newton method. One starts with a chosen $p_0^{(0)}$, and iterates as follows:

$$\left(p_0^{(j+1)}\right)^T = \left(p_0^{(j)}\right)^T - \left(\nabla F(p_0^{(j)})\right)^T \left(\nabla^2 F(p_0^{(j)})\right)^{-1}, \quad (23)$$

where ∇F and $\nabla^2 F$ are the gradient and Hessian of F with respect to p_0 and are given by

$$\begin{aligned} \nabla F(T, p_0^{(j)}) &= \left(\frac{\partial z}{\partial p_0}(T, p_0^{(j)})\right)^T z(T, p_0^{(j)}), \\ \nabla^2 F(T, p_0^{(j)}) &= \left(\frac{\partial z}{\partial p_0}(T, p_0^{(j)})\right)^T \left(\frac{\partial z}{\partial p_0}(T, p_0^{(j)})\right) + z^T(T, p_0^{(j)}) \frac{\partial^2 z}{\partial p_0^2}(T, p_0^{(j)}). \end{aligned}$$

For a sufficiently close initial iterate, the term $\partial^2 z / \partial p_0^2$ is assumed very small, so it is ignored. This means that (23) becomes

$$\left(p_0^{(j+1)}\right)^T = \left(p_0^{(j)}\right)^T - z^T \frac{\partial z}{\partial p_0} \left(\frac{\partial z^T}{\partial p_0} \frac{\partial z}{\partial p_0} \right)^{-1}.$$

This iteration process requires both z and $\partial z/\partial p_0$ at the final time T . The value of z can be found simply by solving the Hamiltonian system (20) forward to T , starting from the initial condition $z_0 = (x_0, p_0)^T$. The partial derivatives can be found in the following way. If the Hamiltonian system is $\dot{z} = \overline{H}z + h(z)$, with $z_0 = (x_0, p_0)^T$, then differentiating with respect to p_0 produces

$$\begin{cases} \frac{d}{dt} \frac{\partial z}{\partial p_0} &= \left(\overline{H} + \frac{\partial h}{\partial z}(z) \right) \frac{\partial z}{\partial p_0} \\ \frac{\partial z}{\partial p_0}(0) &= \begin{pmatrix} 0 \\ I \end{pmatrix}. \end{cases}$$

Adjoining this to the Hamiltonian system results in a $2m(m+1)$ dimensional system of ODEs to be solved on $[0, T]$.

In the method of Markman and Katz the minimization of $F(T, p_0)$ over p_0 is actually carried out for a sequence of time values T increasing toward infinity. This results in a sequence of values for p_0 which converge on the final value as the finite-time problems get closer to the actual infinite-time problem. The solution of the Hamiltonian system generated by the final p_0 in the sequence is then used to obtain the desired control $u(t)$. This process is important because values of p_0 early in the sequence may not be very good, and may cause the system to blow up if T is chosen too large. Starting with a small T and increasing it from there helps to avoid this. It is also useful in showing that the values of p_0 which are found actually are converging as $T \rightarrow \infty$. The TPBV method described in Section 3.4 does not need to be solved for a sequence of final times, because there is no threat of the problem blowing up. However, it could be useful (though time-consuming) to do this to make certain that T has not been chosen too small to give a good approximation of the infinite-time problem.

The initial value of p_0 can be chosen from the linear quadratic regulator problem. That is, $p_0^{(0)} = 2Px_0$ where P is the solution to the algebraic Riccati equation. For subsequent values of T in the sequence the previous final iterate can be used as the new initial iterate. The interpolation of the open-loop controls can be carried out in the same way as described in the previous section, using the specified Green's functions with $\alpha = 3.7$.

As with the previous method, the interpolation makes the calculation of this feedback control a very time-consuming process. However, the iterative method of finding the open-loop control should be somewhat faster than the discretized TPBV problem as the problem dimension increases. It can also be refined by adding more interpolation points in the same way as the previous method. There is no strict limit on the types of problem to which it can be applied, but as with the SGA method of Section 3.3, the performance of this method depends in an essential way on the choice of the initial iterate.

4. Application to Test Problems

In this section we will compare the performance of the five methods discussed in the previous section on several test examples. We will consider their control authority (i.e., the ability to drive the states to zero), the amount of control used, the overall cost of the methods, and the computational

time and flops (floating point operations) used. All computations were done using codes written in MATLAB, implemented on a Sun Ultra 5.

In implementing Ito and Schroeter's TPBV problem method, the iteration process was run until $\|F\| < 10^{-5}$. The Markman and Katz iterative method was cut off when $\|z_i(T) - z_{i-1}(T)\| < 10^{-5}$, for each T , and $\|z(T_j) - z(T_{j-1})\| < 10^{-5}$ in the sequence of T values. The particular sequence of T values used varies among the test problems, as do the discretization interval Δt and final time T used in the TPBV problem method. Another parameter that varies is the region Ω over which the SGA method integrates and the TPBV and Iterative methods interpolate (the number of interpolation points used varies as well).

4.1. Example 1

The first example is described by Cloutier, D'Souza, and Mracek in [32], and contains two state variables and two control variables. The system, which has cubic nonlinearities in each equation, is given by

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} x_1 - x_1^3 + x_2 \\ x_1 + x_1^2 x_2 - x_2 \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}.$$

We factor the nonlinear function $f(x) = A(x)x$ by

$$f(x) = \begin{bmatrix} 1 - x_1^2 & 1 \\ 1 & x_1^2 - 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

for the SDRE method. For the problem, we used the initial condition $x_0 = (1, 1)^T$. The cost functional to be minimized is

$$J(x_0, u) = \int_0^\infty \left(x^T \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{bmatrix} x + u^T \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} u \right) dt.$$

For the TPBV method, our calculations used $\Delta t = 0.05$ and the final time $T = 5.25$. For the iterative method, the sequence of T values was started at $T_0 = 1.0$ and increased by equally spaced time intervals of $\Delta T = 0.25$. The region $\Omega = [-1, 1] \times [-1, 1]$, and $100 = 10 \times 10$ regularly spaced interpolation points were used.

Figure 1 shows $\|x\|$ for the five feedback controls described earlier, and also for the open-loop control found using the discretized two point boundary value problem approach for the initial condition x_0 . All of these results are extremely close to each other, with the TPBV based method performing slightly better, and the SGA method performing slightly worse. The norms of the control $\|u\|$ are shown in Figure 2 for all five methods. There are only small differences between most of the methods, with the clear exception being the SGA method, which requires nearly twice as much control initially.

The time and flops required for each feedback control, along with each of their values for the cost functional $J(x_0, u)$, are listed in Table 1. To compare, the open-loop control cost is 4.7965, and the linear control (which also succeeds in driving the problem to zero in this example) cost is 4.7810. The costs are all very similar (except SGA), varying by only about 0.5% among themselves, at about 5% less than the linear control. The computational time for TPBV and Iter, involving the solution

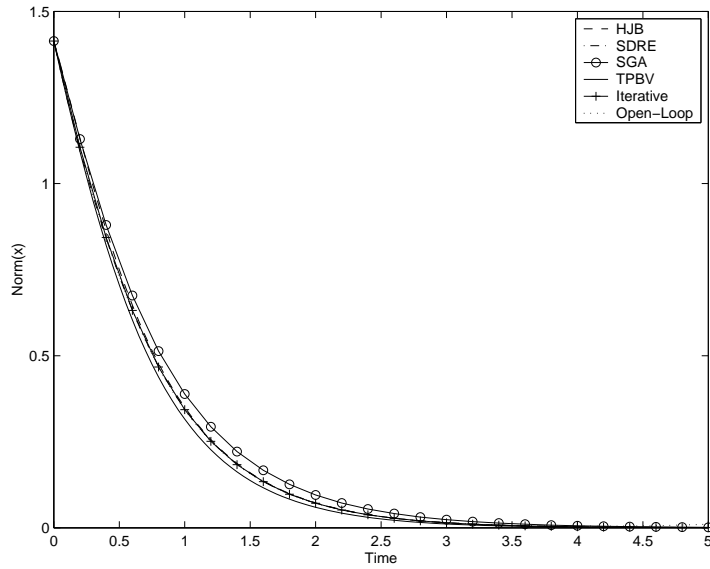


Figure 1: Comparison of the norms of feedback controlled trajectories in Example 1.

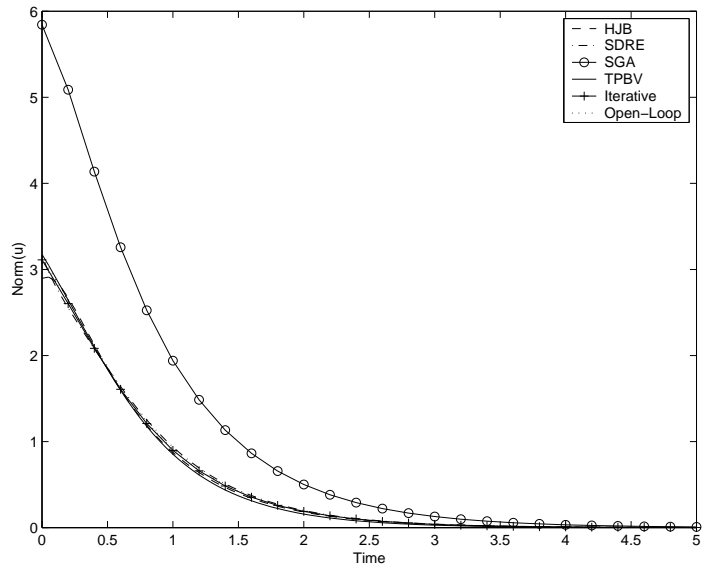


Figure 2: Comparison of the norms of feedback controls in Example 1.

of many open-loop problems, is much larger than that of HJB and SDRE. The SGA computational time is in between these extremes, but is still much less than the interpolation methods. However, the expensive computations of SGA, TPBV and Iter are all done off-line, so they could be run again very quickly with a different x_0 if so desired. For this problem the best choice of methodology appears

	CPU Time (sec.)	Flops	Cost
HJB	0.15	4.71×10^4	4.6985
SDRE	0.21	4.89×10^4	4.6929
SGA	1.52	1.69×10^6	17.5309
TPBV	448.46	1.62×10^{10}	4.6809
Iter	974.89	2.28×10^8	4.6768

Table 1: Numerical comparison of feedback control methodologies in Example 1.

to be HJB or SDRE: they achieve results virtually as good as any other method, and do so with minimal computation.

4.2. Example 2

The second example we chose to test the various feedback control methods is from Markman and Katz [20]. It has two state variables and one control variable, and has a cubic nonlinearity in the system, which is defined by

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ x_1^3 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u.$$

The function $f(x)$ is rewritten as

$$f(x) = \begin{bmatrix} 0 & 1 \\ x_1^2 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

for the SDRE method. The initial condition is $x_0 = (1, 0)^T$ and the cost functional to be minimized is

$$J(x_0, u) = \int_0^\infty \left(x^T \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{bmatrix} x + \frac{1}{2} u^2 \right) dt.$$

For this example, in the TPBV method we used $\Delta t = 0.05$ and $T = 7.50$. For the iterative method, $T_0 = 1.0$ and $\Delta T = 0.5$. The same region $\Omega = [-1, 1] \times [-1, 1]$ as in Example 1 was used, with $100 = 10 \times 10$ interpolation points.

Figure 3 depicts the state x_1 and Figure 4 displays the control u for all five control methods. The SGA method iterations never actually converged to a final control; the one shown here (which does force the system to zero) is taken after an iteration number truncation. It can be seen in the figures that the interpolation methods drive the states to zero slightly more slowly than the SDRE and HJB methods, but the SDRE and HJB methods require more control to achieve this.

Table 2 contains the time and flops required and cost for each control method. To compare, the open-loop control cost is 1.5624. For this example the linear control fails to bring the system to zero. The costs are similar (again with the exception of SGA), but not as close as for Example 1. They vary by about 5% among themselves, and are effective when the linear control fails. The

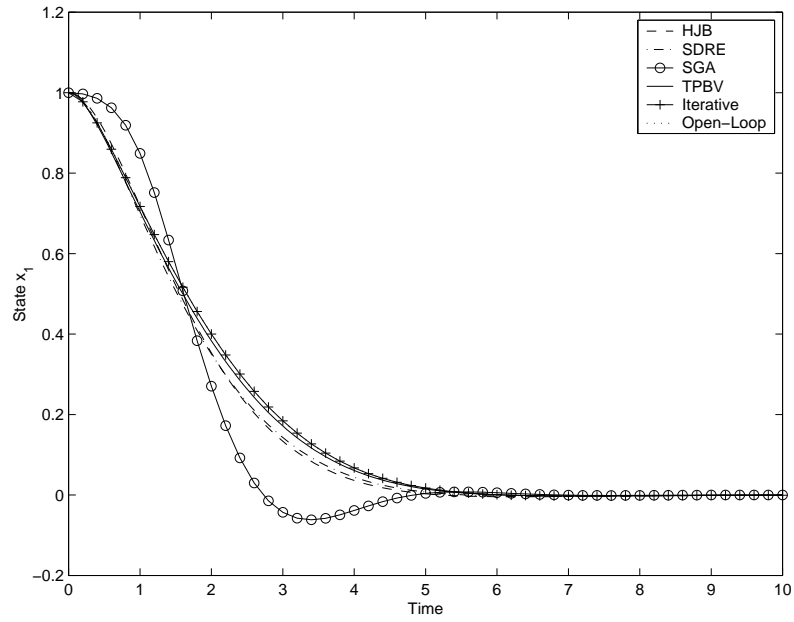


Figure 3: Comparison of feedback controlled states x_1 in Example 2.

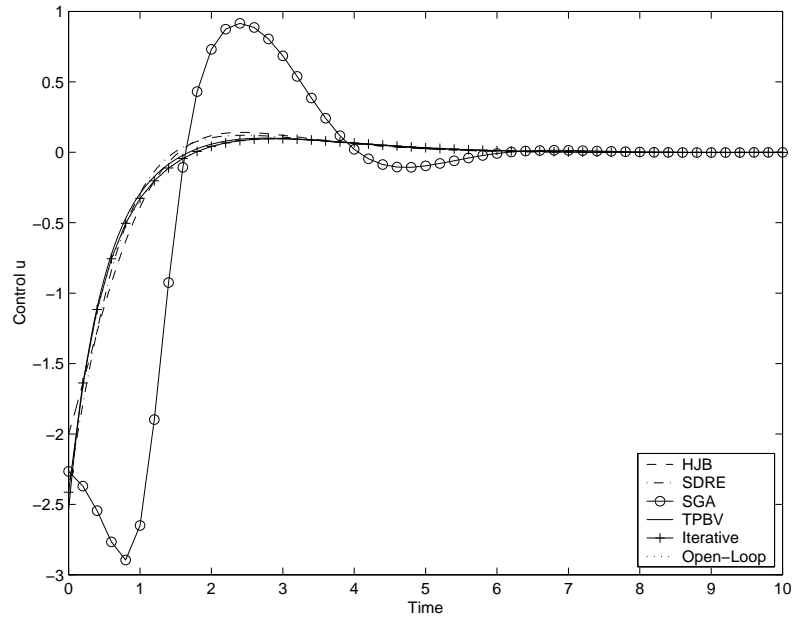


Figure 4: Comparison of feedback controls in Example 2.

	CPU Time (sec.)	Flops	Cost
HJB	0.25	8.43×10^4	1.5028
SDRE	0.30	9.26×10^4	1.5629
SGA	2.19	2.10×10^6	5.5935
TPBV	1413.3	4.90×10^{10}	1.4867
Iter	1065.9	2.47×10^8	1.4842

Table 2: Numerical comparison of feedback control methodologies in Example 2.

computational times are very much like those in Example 1, with the times for TPBV and Iter again much larger than those for HJB and SDRE, and SGA in between. The best choice of control here might be HJB: its performance is almost as good as the interpolation methods, without the intensive computations, and its cost is less than that of SDRE.

With this example we also tried using different initial conditions to see how the various controls would perform with starting points farther away from the origin. We test both $x_0 = (2, 0)^T$ and $x_0 = (4, 0)^T$. Note that both of these points are outside the region Ω used for the interpolations in TPBV and Iter and the integrations in SGA. Thus, some loss of effectiveness in these methods should be expected. The HJB and SDRE methods should also suffer, since their power series approximations will be less accurate farther away from the expansion point of the origin.

It turns out that the SDRE and SAG methods both fail to stabilize the system, for each of these initial conditions. The value of the cost functional J for the other three methods (and the open-loop control calculated through the TPBV problem), for both initial conditions, are displayed in Table 3. The state x_1 and control u for $x_0 = (4, 0)^T$ are graphed in Figures 5 and 6. The results for

	Cost for $(2, 0)^T$	Cost for $(4, 0)^T$
HJB	28.19	1340
TPBV	25.73	2170
Iter	26.95	5720
Open-Loop	27.66	982

Table 3: Feedback control methodologies in Example 2 with a distant initial state.

$x_0 = (2, 0)^T$ are not too different from those for $x_0 = (1, 0)^T$, at least for those methods that are still effective: the cost differences between the three feedback controls and the open-loop control expand somewhat, up to nearly a 10% difference between them, and the open-loop control is relatively more effective than before. However, for $x_0 = (4, 0)^T$ there are dramatic differences. The open-loop control is much more effective, which is expected since it is being solved for this point specifically rather than depending on an interpolation region or power series expansion point which is far away. The HJB method is also shown to be much better than TPBV (which is in turn much better than Iter), in terms of stabilization, amount of control, and cost. For this example at least, the HJB method produces the feedback control least affected by the choice of an initial state away from the origin.

Since the two-term HJB method performed very well for this example, we also calculated more terms in the power series to see how much is lost by restricting it to two terms. This is complicated, as

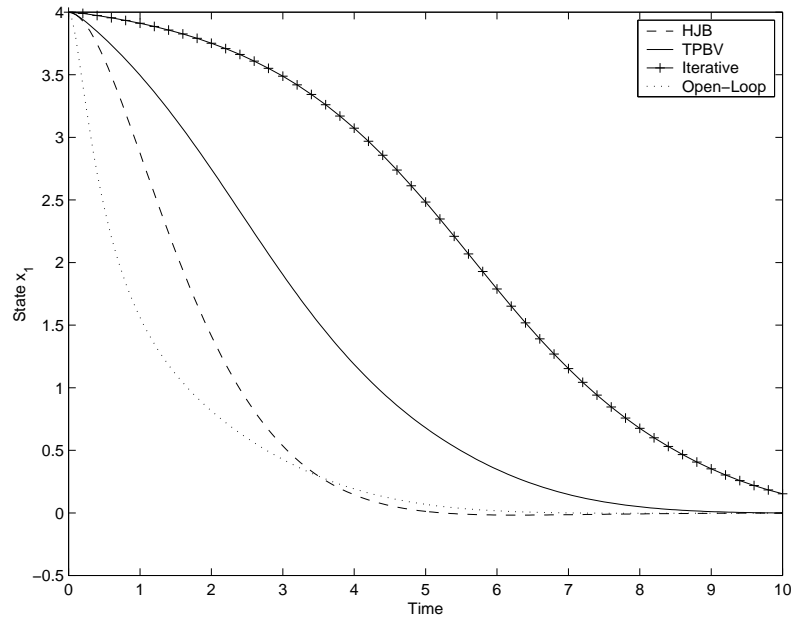


Figure 5: Feedback controlled states x_1 in Example 2 with a distant initial state.

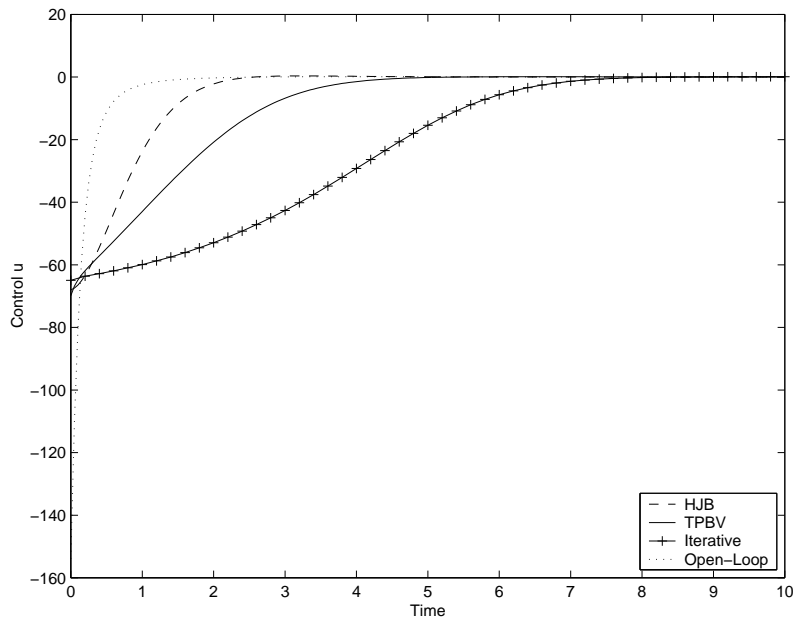


Figure 6: Feedback controls in Example 2 with a distant initial state.

mentioned in Section 3.1, but less so for this example since the system is relatively simple. Fifth and seventh order terms were added to the first and third order terms already found, and the resulting control was used on the example with initial condition $x_0 = (1, 0)^T$. The result was slightly better than the two-term control in its effect on the state variables, but used slightly more control and resulted in a cost of 1.5060 compared to the two-term control's cost of 1.5028. For this example, we can conclude that very little is lost by the restriction to two terms.

4.3. Example 3

The third example we studied is a model of the flight dynamics of a high-performance aircraft taken from Garrard, Enns and Snell [5]. This is a higher dimension problem than the first two, containing five state variables and one control variable. It has quadratic nonlinearities only. We have changed the problem from the description in the paper by scaling the variables so that they are closer to the same order. This was done by converting from degrees to radians in x_5 and u in order to match the units of the other variables, and by changing x_1 from m/s to (100m)/s. We also changed the cost functional to one of the form we are considering (quadratic in the state and control).

The state variables in this model represent the flight conditions of the aircraft: x_1 is the deviation of the velocity from the level flight trim value of 1(100m)/s (given in units of (100m)/s), x_2 is the deviation of the angle of attack (the angle of the aircraft wing with respect to the direction of the velocity) from the trim value of $4.2(\pi/180)$ radians, x_3 is the pitch rate (the rate of change in the angle of the aircraft with respect to horizontal) in rad/s, x_4 is the flight path angle (the angle of the velocity with respect to horizontal) in radians, and x_5 is the deviation of the canard deflection angle in radians from the trim value, which is not given. The control u is the input canard deflection in radians. The canards are control flaps which can deflect downward by up to $90(\pi/180)$ radians. There is a lag in the input given to them included in the model. The initial condition chosen for this example is a high angle of attack (x_2) with otherwise trim values; the control methods will attempt to stabilize the aircraft by driving all the variables to the trim conditions. (The meaning of the model is described in more detail in [5] for interested readers.)

The system is given by

$$\dot{x} = (A_0 + x_2 A_{NL})x + Bu,$$

where the matrices A_0 , A_{NL} and B are all constant-valued, and are given by:

$$A_0 = \begin{bmatrix} -0.0443 & 1.1280 & 0.0 & -0.0981 & 0.0 \\ -0.0490 & -2.5390 & 1.0 & 0.0 & -0.0854 \\ -0.0730 & 19.3200 & -2.2700 & 0.0 & 22.6834 \\ 0.0490 & 2.5390 & 0.0 & 0.0 & 0.0854 \\ 0.0 & 0.0 & 0.0 & 0.0 & 20.0 \end{bmatrix},$$

$$A_{NL} = \begin{bmatrix} -0.2317 & 0.0 & 0.0 & 0.0 & 0.0 \\ -1.2760 & -0.7922 & 0.0 & 0.0 & 0.0206 \\ 0.1020 & 64.2940 & -13.9710 & 0.0 & -5.4167 \\ 1.2760 & 0.7922 & 0.0 & 0.0 & -0.0206 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \end{bmatrix},$$

$$B = \begin{bmatrix} 0.0 & 0.0 & 0.0 & 0.0 & 20.0 \end{bmatrix}^T.$$

The initial condition is $x_0 = (0, 25(\pi/180), 0, 0, 0)^T$. The cost functional to be minimized is

$$J(x_0, u) = \int_0^\infty (x^T (I_5) x + 100u^2) dt,$$

where the matrix $Q = I_5$ is a 5×5 identity matrix. For this example, we used $\Delta t = 0.15$ and $T = 15.00$, and $T_0 = 0.5$ and $\Delta T = 0.25$. The region used here is $\Omega = [-0.3, 0.3] \times [-25(\pi/180), 25(\pi/180)]^4$, with $1024 = 4^5$ interpolation points.

This example appears to be much more difficult to solve than the previous ones. This may be caused by the larger dimension of the problem, or because the system dynamics are more highly unstable than the previous examples, or because the initial state we chose was far enough away from zero to make the problem very difficult. The SGA method does not converge. The Iter algorithm for finding the open-loop control for a given initial point fails to converge for many of the points in the interpolation domain. The TPBV method does find the open-loop controls for all the interpolation points, but the feedback control created by the interpolation fails to drive the system to zero.

Figure 7 depicts the state x_2 , which is the primary variable we want to control and the one which is initially nonzero. Figure 8 displays the norm of the entire state vector, $\|x\|$, to provide a better

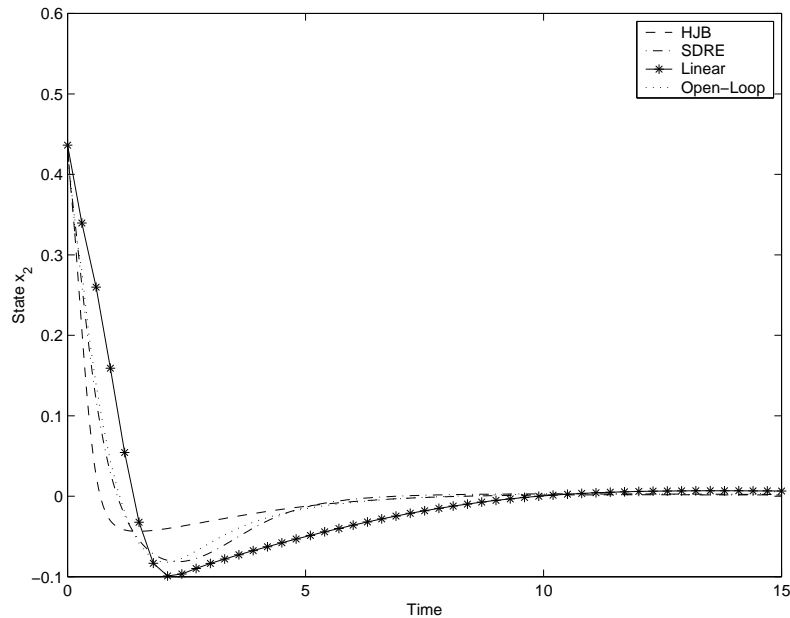


Figure 7: Comparison of feedback controlled states x_2 in Example 3.

view of the overall effect of the control methods on the system. Figure 9 depicts the control u . These graphs are all done for HJB, SDRE, the open-loop control found by the TPBV problem discretization, and the linear control, all of which successfully stabilized the system. The HJB method performs better on both the state x_2 and the norm $\|x\|$ overall. SDRE performs somewhat better than the open-loop control with $\|x\|$ but slightly worse with x_2 . All three methods are more effective than the linear control in terms of stabilization. In comparison, the control used by the HJB method is larger than SDRE for the initial burst but then lower after about 0.5. Similarly, SDRE uses more

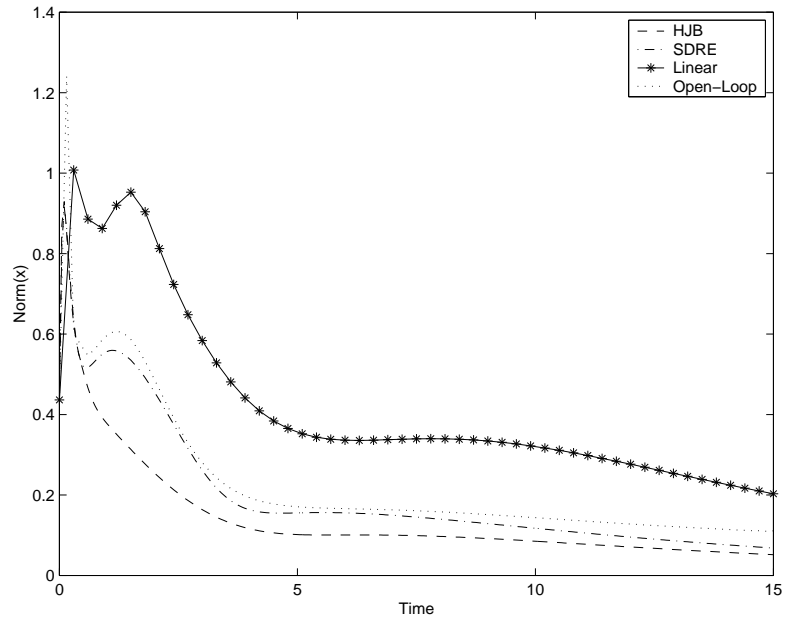


Figure 8: Comparison of the norms of feedback controlled trajectories in Example 3.

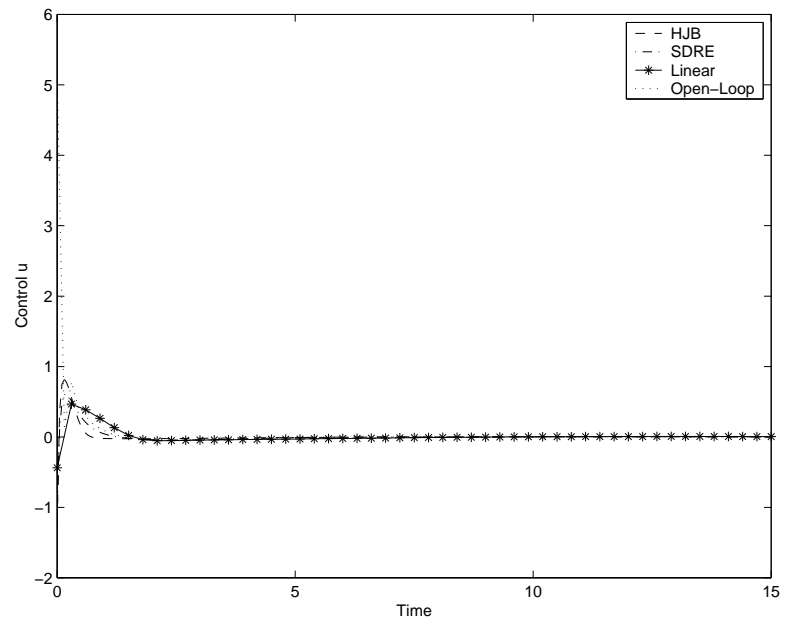


Figure 9: Comparison of feedback controls in Example 3.

control than the linear control in the initial burst but is lower thereafter. The open-loop control is very large initially before dropping down to the level of the other controls. This may explain why the interpolation of these open-loop TPBV controls failed to produce an effective closed-loop control. The interpolation is carried out on the initial control, which in this case is very large in comparison to the feedback controls and even the open-loop control later in time. This may make it difficult to extend with an interpolation, especially when it is done so roughly (with only four points in each dimension).

The time and flops required and cost functional $J(x_0, u)$ for the two successful nonlinear feedback controls are listed in Table 4. The open-loop control cost for this problem is 107.58, very large

	CPU Time (sec.)	Flops	Cost
HJB	0.88	1.44×10^6	13.7983
SDRE	1.10	1.09×10^6	11.3175

Table 4: Numerical comparison of feedback control methodologies in Example 3.

compared with the others due to the large initial control which is weighted heavily in the cost functional. Interestingly, the linear control for this example has a cost of only 10.2013, apparently because it has very little of the heavily-weighted control compared with the other methods. The stabilization effect of the linear control is smaller compared with the other methods, however, as shown in Figures 7 and 8.

Determining which method is best is more difficult here than for the first two examples. The SDRE and linear control have lower costs but less effectiveness in stabilizing the state variables, while the HJB control appears better in terms of state stabilization but results in a higher cost. The larger weight on the control in the cost functional causes this higher cost, by amplifying the small differences in the controls seen in Figure 9 into large cost differences. The choice of the large weight in the cost functional was made to avoid nonphysical results such as the huge initial spike seen in the open-loop control. This was successful for the other methods, but it also results in the HJB method having a higher cost even though it uses only slightly more control. So even though the linear control has the lowest cost, the HJB could well be considered the best choice for this example, based on its effect on the states. This is unlike in Example 1, where the cost and the state behavior match up well, and in Example 2, where the effect on the cost functional from the control was based on a greater difference between the methods.

4.4. Example 4

This example is taken from Garrard and Jordan [4], and is a different model of flight control. It has only three state variables instead of five but contains both quadratic and cubic nonlinear terms. In this model the state variables are as follows: x_1 is the deviation of the angle of attack from the trim value of 0.044 radians, x_2 is the flight path angle in radians, and x_3 is the rate of change of the flight path angle in rad/s. The control u is the deviation of the tail deflection angle (with respect to the wing direction) from the trim value of -0.009 radians. In the aircraft described by this model, the entire tail can rotate to control its flight. As in the previous example, the control methods are given an initial condition of a high angle of attack (x_1) but otherwise trim conditions, and are asked

to stabilize the system. (The interested reader can find more information on this model in [4].)

The system is written as

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} -0.877x_1 + x_3 + 0.47x_1^2 - 0.088x_1x_3 - 0.019x_2^2 + 3.846x_1^3 - x_1^2x_3 \\ x_3 \\ -4.208x_1 - 0.396x_3 - 0.47x_1^2 - 3.564x_1^3 \end{bmatrix} + \begin{bmatrix} -0.215 \\ 0.0 \\ -20.967 \end{bmatrix} u$$

with an initial condition of $x_0 = (25(\pi/180), 0, 0)^T$. For the SDRE method the nonlinear function is rewritten as

$$f(x) = \begin{bmatrix} -0.877 + 0.47x_1 + 3.846x_1^2 & -0.019x_2 & 1 - 0.088x_1 - x_1^2 \\ 0 & 0 & 1 \\ -4.208 - 0.47x_1 - 3.564x_1^2 & 0 & -0.396 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}.$$

The cost functional is

$$J(x_0, u) = \int_0^\infty \left(x^T \left(\frac{1}{4} I_3 \right) x + u^2 \right) dt.$$

For this example, in the TPBV method we used $\Delta t = 0.1$ and $T = 12.0$, and in the Iterative method $T_0 = 0.5$ and $\Delta T = 0.25$. The region $\Omega = [-25(\pi/180), 25(\pi/180)]^3$ was used and there are $125 = 5^3$ interpolation points.

As with the other flight problem in Example 3, neither of the iterative methods converge. The TPBV method, which failed for that problem, does work here. Figure 10 depicts the state x_1 and Figure 11 displays the control u for HJB, SDRE, TPBV, and the open-loop control found by the TPBV problem discretization. The TPBV method is clearly the best at controlling x_1 , with the open-loop control a close second. Both of these have a large initial control which quickly becomes very small. In contrast, the SDRE and HJB methods use more constant control, with less control than the other methods at first but more after 0.5 or so. They are also significantly worse at controlling x_1 , with the HJB method the worst of the four. This should be expected, since this is our first example with a relatively complicated nonlinear structure. The two-term HJB method ignores the cubic terms in the system, as explained in Section 3.1, only using the linear and quadratic parts. The nonlinear part of the system cannot be rewritten for the SDRE method as $\Delta A = g(x)\Delta A_C$ (as described in Section 3.2), so the power series is difficult to compute for that method. Only two terms were used in this example, compared with five terms for the other problems. The TPBV method does not have such strong drawbacks with this type of problem, so it is naturally much more effective than the other feedback controls.

The time, flops and value of the cost functional $J(x_0, u)$ for each control method are displayed in Table 5. The open-loop control has a cost of 0.0483, and the linear control has a cost of 0.0576.

	CPU Time (sec.)	Flops	Cost
HJB	0.42	2.29×10^5	0.0856
SDRE	0.41	1.85×10^5	0.0578
TPBV	6014	2.14×10^{11}	0.0435

Table 5: Numerical comparison of feedback control methodologies in Example 4.

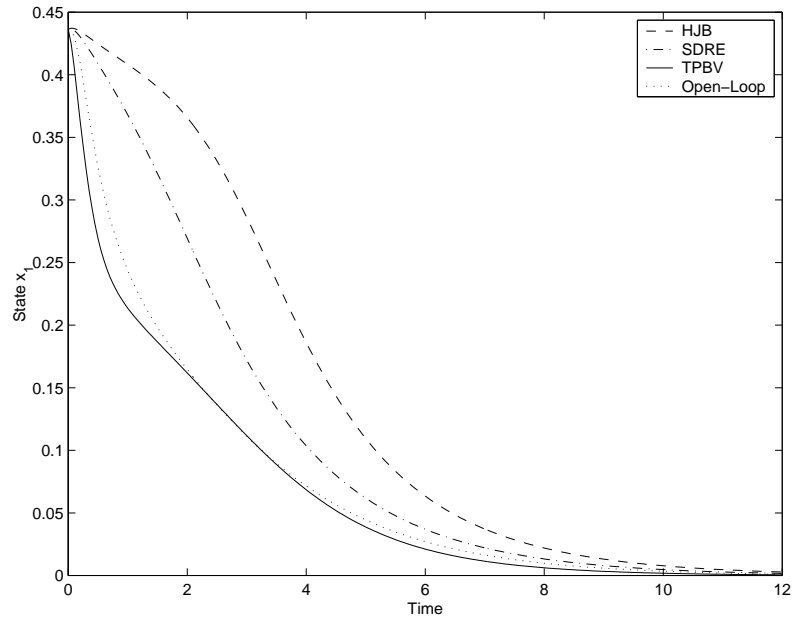


Figure 10: Comparison of feedback controlled states x_1 in Example 4.

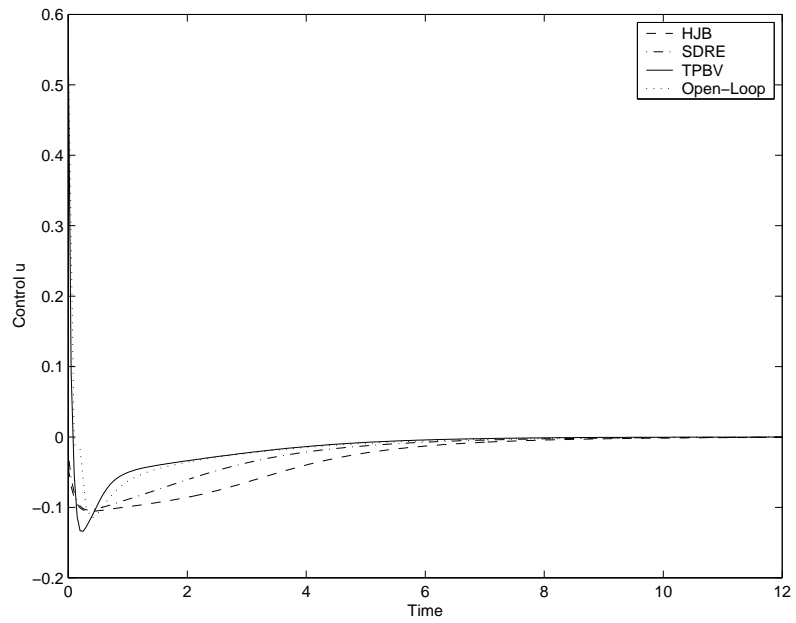


Figure 11: Comparison of feedback controls in Example 4.

These match the results seen in Figures 10 and 11, with the TPBV slightly better than the open-loop, with SDRE significantly less effective and HJB clearly last in performance. The TPBV interpolation method takes a very long time to do all the open-loop calculations it needs, but for this example it is well worth the effort since it produces much better results than the other feedback controls.

5. Conclusions

In comparing several feedback control methods for nonlinear systems, we have seen that for many problems it appears that a simple method is the best choice. The method of Garrard using only two power series terms is very effective for problems with one level of nonlinearity, a category including many problems of great interest. For Examples 1-3, which are all of this type, the two-term HJB method produced results nearly as good or better than the other feedback control methods examined. It is very easy to implement and does not require the much larger computational time some of the other methods need. It also proved more effective in Example 2 for initial states farther away from the origin. For problems with more complex nonlinearities, such as Example 4, the two-term HJB method is not very effective, so a less restricted method is a better choice. In this case the TPBV interpolation method produced good results, although requiring a great deal of computational time. The more complicated iterative methods had difficulties with some of the examples, failing to converge because of inadequate initial iterates. Convergence results are given in [10, 21] for the iterative methods, but these require restrictions, for example on the initial iterate, which do not hold for our examples.

References

- [1] Anderson, B. D. O., and Moore, J. B., *Optimal Control: Linear Quadratic Methods*, Englewood Cliffs, New Jersey: Prentice-Hall, 1990.
- [2] Lewis, F. L., and Syrmos, V. L., *Optimal Control*, New York: Wiley, 1995.
- [3] Garrard, W. L., "Suboptimal Feedback Control for Nonlinear Systems," *Automatica*, Vol 8, pp219-221 (1972).
- [4] Garrard, W. L., and Jordan, J. M., "Design of Nonlinear Automatic Flight Control Systems," *Automatica*, Vol 13, pp497-505 (1977).
- [5] Garrard, W. L., Enns, D. F., and Snell, S. A., "Nonlinear Feedback Control of Highly Manoeuvrable Aircraft," *International Journal of Control*, Vol 56, pp799-812 (1992).
- [6] Nishikawa, Y., Sannomiya, N., and Itakura, H., "A Method for Suboptimal Design of Nonlinear Feedback Systems," *Automatica*, Vol 7, pp703-712 (1971).
- [7] Leake, R. J. and Liu, R. W., "Construction of Suboptimal Control Sequences," *SIAM Journal on Control and Optimization*, Vol 5, pp54-63 (1967).
- [8] Saridis, G. N., and Lee, C. S. G., "An Approximation Theory of Optimal Control for Trainable Manipulators," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol 9, pp152-159 (1979).
- [9] Beard, R. W., Saridis, G. N., and Wen, J. T., "Galerkin Approximation of the Generalized Hamilton-Jacobi-Bellman Equation," *Automatica*, Vol 33, pp2159-2177 (1997).
- [10] Beard, R. W., Saridis, G. N., and Wen, J. T., "Approximate Solutions to the Time-Invariant Hamilton-Jacobi-Bellman Equation," *Journal of Optimization Theory and Applications*, Vol 96, pp589-626 (1998).
- [11] Wernli, A. and Cook, G., "Suboptimal Control for the Nonlinear Quadratic Regulator Problem," *Automatica*, Vol 11, pp75-84 (1975).
- [12] Krikelis, N. J. and Kiriakidis, K. I., "Optimal Feedback Control of Non-linear Systems," *International Journal of Systems Science*, Vol 23, pp2141-2153 (1992).
- [13] Cloutier, J. R., D'Souza, C. N., and Mracek, C. P., "Nonlinear Regulation and Nonlinear H_∞ Control Via the State-Dependent Riccati Equation Technique: Part 1. Theory," *Proceedings of the First International Conference on Nonlinear Problems in Aviation and Aerospace*, Daytona Beach, FL, May 1996.
- [14] Halme, A., and Hamalainen, R. P., "On the Nonlinear Regulator Problem," *Journal of Optimization Theory and Applications*, Vol 16, pp255-275 (1975).
- [15] Goh, C. J., "On the Nonlinear Optimal Regulator Problem," *Automatica*, Vol 29, pp751-756 (1993).

- [16] Ito, K. and Schroeter, J. D., "Reduced Order Feedback Synthesis for Viscous Incompressible Flows," Technical Report CRSC-TR98-41, Center for Research in Scientific Computation, North Carolina State University (1998).
- [17] Kirk, D. E., *Optimal Control Theory*, Englewood Cliffs, New Jersey: Prentice-Hall, 1970.
- [18] Sage, A. P., and White, C. C., III, *Optimum Systems Control, 2nd Edition*, Englewood Cliffs, New Jersey: Prentice-Hall, 1977.
- [19] Bosarge, W. E., Johnson, O. G., McKnight, R. S., and Timlake, W. P., "The Ritz-Galerkin Procedure for Nonlinear Control Problems," *SIAM Journal on Numerical Analysis*, Vol 10, pp94-110 (1973).
- [20] Markman, J. and Katz, I. N., "An Iterative Algorithm for Solving Hamilton Jacobi Type Equations," not yet published.
- [21] Markman, J. and Katz, I. N., "Convergence of an Iterative Algorithm for Solving Hamilton Jacobi Type Equations," not yet published.
- [22] Hofer, E. P. and Tibken, B., "An Iterative Method for the Finite-Time Bilinear Quadratic Control Problem," *Journal of Optimization Theory and Applications*, Vol 57, pp411-427 (1988).
- [23] Aganovic, Z. and Gajic, Z., "The Successive Approximation Procedure for Finite-Time Optimal Control of Bilinear Systems," *IEEE Transactions on Automatic Control*, Vol 29, pp1932-1935 (1994).
- [24] Cebuhar, W. A. and Costanza, V., "Approximation Procedures for the Optimal Control of Bilinear and Nonlinear Systems," *Journal of Optimization Theory and Applications*, Vol 43, pp615-627 (1984).
- [25] Ryan, E. P., "Optimal Feedback Control of Bilinear Systems," *Journal of Optimization Theory and Applications*, Vol 44, pp333-362 (1984).
- [26] Tzafestas, S. G., Anagnostou, K. E., and Pimenides, T. G., "Stabilizing Optimal Control of Bilinear Systems with a Generalized Cost," *Optimal Control Applications and Methods*, Vol 5, pp111-117 (1984).
- [27] Werner, R. A. and Cruz, J. B., "Feedback Control Which Preserves Optimality for Systems with Unknown Parameters," *IEEE Transactions on Automatic Control*, Vol 13, pp621-629 (1968).
- [28] Qu, Z., "Robust Control of Nonlinear Uncertain Systems without Generalized Matching Conditions," *IEEE Transactions on Automatic Control*, Vol 40, pp1453-1460 (1995).
- [29] Bourdache-Siguerdidjane, H. and Fliess, M., "Optimal Control of Non-linear Systems," *Automatica*, Vol 23, pp373-379 (1987).
- [30] Banks, H. T., Smith, R. C., and Wang, Y., *Smart Material Structures: Modeling, Estimation and Control*, Chichester, England: Wiley, 1996.
- [31] Lasiecka, I. and Triggiani, R., *Differential and Algebraic Riccati Equations with Application to Boundary/Point Control Problems*, New York: Springer-Verlag, 1991.

- [32] Cloutier, J. R., D'Souza, C. N., and Mracek, C. P., "Nonlinear Regulation and Nonlinear H_∞ Control Via the State-Dependent Riccati Equation Technique: Part 2. Examples," Proceedings of the First International Conference on Nonlinear Problems in Aviation and Aerospace, Daytona Beach, FL, May 1996.