



## Faculty Publications

---

1998-11

# Successive Galerkin Approximation Algorithms for Nonlinear Optimal and Robust Control

Timothy McLain

*Mechanical Engineering Department, Brigham Young University, mclain@byu.edu*

Randal W. Beard

*Department of Electrical Engineering, Brigham Young University, beard@ee.byu.edu*

Follow this and additional works at: <https://scholarsarchive.byu.edu/facpub>



Part of the [Mechanical Engineering Commons](#)

## Original Publication Citation

Beard, R. and McLain, T. Successive Galerkin Approximation Algorithms for Nonlinear Optimal and Robust Control, *International Journal of Control: Special Issue on Breakthroughs in the Control of Nonlinear Systems*, vol. 71, no. 5, pp. 717-743, November 1998.

---

## BYU ScholarsArchive Citation

McLain, Timothy and Beard, Randal W., "Successive Galerkin Approximation Algorithms for Nonlinear Optimal and Robust Control" (1998). *Faculty Publications*. 1910.  
<https://scholarsarchive.byu.edu/facpub/1910>

This Peer-Reviewed Article is brought to you for free and open access by BYU ScholarsArchive. It has been accepted for inclusion in Faculty Publications by an authorized administrator of BYU ScholarsArchive. For more information, please contact [ellen\\_amatangelo@byu.edu](mailto:ellen_amatangelo@byu.edu).

## Successive Galerkin approximation algorithms for nonlinear optimal and robust control

RANDAL W. BEARD<sup>†</sup> and TIMOTHY W. McLAIN<sup>‡</sup>

Nonlinear optimal control and nonlinear  $\mathcal{H}_\infty$  control are two of the most significant paradigms in nonlinear systems theory. Unfortunately, these problems require the solution of Hamilton–Jacobi equations, which are extremely difficult to solve in practice. To make matters worse, approximation techniques for these equations are inherently prone to the so-called ‘curse of dimensionality’. While there have been many attempts to approximate these equations, solutions resulting in closed-loop control with well-defined stability and robustness have remained elusive. This paper describes a recent breakthrough in approximating the Hamilton–Jacobi–Bellman and Hamilton–Jacobi–Isaacs equations. Successive approximation and Galerkin approximation methods are combined to derive a novel algorithm that produces stabilizing, closed-loop control laws with well-defined stability regions. In addition, we show how the structure of the algorithm can be exploited to reduce the amount of computation from exponential to polynomial growth in the dimension of the state space. The algorithms are illustrated with several examples.

### 1. Introduction

Two important paradigms in linear control theory are  $\mathcal{H}_2$  optimal control and  $\mathcal{H}_\infty$  robust control (Doyle *et al.* 1988). Both paradigms extend naturally to nonlinear systems. Although the spaces  $\mathcal{H}_2$  and  $\mathcal{H}_\infty$  are not used in the nonlinear setting, we follow the current nomenclature in the literature and refer to the corresponding nonlinear problems as nonlinear  $\mathcal{H}_2$  control and nonlinear  $\mathcal{H}_\infty$  control. Theoretical results in nonlinear  $\mathcal{H}_2$  control were developed in the late 1950s and 1960s and are contained in numerous textbooks (e.g. Athans and Falb 1966, Fleming and Rishel 1975, Lewis 1986). The development of nonlinear  $\mathcal{H}_\infty$  control is more recent, some of the seminal papers being van der Schaft (1992), Ball *et al.* (1993), and Isidori and Astolfi (1992).

For linear, state-feedback systems, both the  $\mathcal{H}_2$  and  $\mathcal{H}_\infty$  control problems are solved by the solution to a matrix Riccati equation (Bittani *et al.* 1991). Since efficient numerical techniques exist for solving Riccati equations, the  $\mathcal{H}_2$  and  $\mathcal{H}_\infty$  paradigms have been applied to a wide variety of problems, including robotics, flight control and satellite attitude control.

Although the theory for nonlinear  $\mathcal{H}_2$  and  $\mathcal{H}_\infty$  is well developed, there have been very few applications due to the lack of efficient numerical methods for solving these problems. Solving the nonlinear, state-feedback  $\mathcal{H}_2$  optimal control problem requires the solution of the Hamilton–Jacobi–Bellman (HJB) equation (Lewis 1986). Solv-

---

<sup>†</sup> Department of Electrical and Computer Engineering, Brigham Young University, Provo, UT 84602, USA. e-mail: beard@ee.byu.edu

<sup>‡</sup> Department of Mechanical Engineering, Brigham Young University, Provo, UT 84602, USA. e-mail: tmclain@et.byu.edu

ing the nonlinear, state-feedback  $\mathcal{H}_\infty$  control problem requires the solution of the Hamilton–Jacobi–Isaacs (HJI) equation (van der Schaft 1992).

Both the HJB and HJI equations are first-order, nonlinear partial differential equations that have proven to be impossible to solve in general and very difficult to solve for specific nonlinear systems. Since these equations are difficult to solve analytically, there has been a great deal of research directed toward approximating their solutions. This research can be divided into four different categories:

- the method of characteristics;
- series approximation;
- regularization; and
- finite difference/finite element approximation.

The most straightforward use of the method of characteristics is to integrate the solution of the problem forward to find the solution for one initial condition. The result is an open-loop control strategy. Examples for the HJB equation are Kirk (1970) and Sage and White III (1977). An example and for the HJI equation is Wise and Sedwick (1994). This method is often extended by using many open-loop trajectories to train a neural network or by using some other scheme that interpolates between open-loop control trajectories (cf. Goh 1993).

Series approximations of the HJB and HJI equations are found by assuming that the system equations are analytic in the state and forming a Taylor-series expansion of the value function. The approximation is formed by finding a finite number of terms in this truncation. Examples for the HJB equation include Garrard and Jordan (1977) and Garrard *et al.* (1992). For the HJI equation see Huang and Lin (1995).

The basic idea behind ‘regularization’ is to add terms in the cost function of the nonlinear  $\mathcal{H}_2$  control which cancel out the nonlinearities in the system. Examples include Ryan (1984), Tzafestas *et al.* (1984), Lu (1993) and Freeman and Kokotovic (1995).

The classical methods of finite difference and finite element solutions have also been used to approximate the Hamilton–Jacobi equation. To make the method rigorous, the viscosity solutions of HJB and HJI equations are approximated. See Crandall *et al.* (1992) for an introduction to viscosity solutions and Kushner (1990) and Fleming and Soner (1993) for applications to control.

There are problems with each of the methods listed above. The method of characteristics results in open-loop control and requires much memory to implement. For the closed-loop extensions it is difficult to say anything about stability or robustness. The difficulty with series approximations is that it is generally hard to solve for high-order terms in the approximation, and it is difficult to guarantee stability for finite truncations of the series. In addition, the stability region of the approximate control is limited by the region of convergence of the underlying Taylor series. This region is impossible to approximate *a priori*. Regularization is problematic, since the original optimal control problem is no longer being addressed. These controls also become difficult to tune, since tuning the gains may have unexpected effects on the outputs of the system. The problem with finite element and finite difference methods is that the computational load and computer memory required for the approximation grow exponentially with the dimension of the state of the system. As Bellman originally noted, a fundamental problem with Hamilton–Jacobi based methods in general is that they all suffer from the curse of dimensionality. In other words, the amount of

computation and/or memory required to implement the method grows exponentially with the size of the state space.

An ideal approximation method would not suffer from any of the disadvantages of the methods discussed above. In particular, it would:

- result in closed-loop control that is easy to implement;
- have guaranteed stability for finite truncations of the approximation;
- guarantee that the approximation error goes to zero as the order of the approximation increases;
- have a well-defined region of the state space where the approximation is guaranteed to work;
- have low run-time computation and memory requirements;
- effectively deal with the curse of dimensionality;
- be easy to tune; and
- give explicit bounds on the approximation error.

In recent years the authors and their colleagues have developed an approximation technique that incorporates all but the last of these desired properties. The underlying idea is rather simple and consists of two steps. First, Bellman's idea of iteration in policy space is used to reduce the HJB or HJI equation to a sequence of linear partial differential equations. Second, Galerkin's approximation method is used, with basis functions defined globally on some compact set, to approximate each of these PDEs. The resulting approximation scheme has several interesting properties:

- The algorithm must be started with an initial stabilizing control. If the order of approximation is large enough, then one interpretation of the algorithm is that it is improving the performance of the control at each iteration.
- The region (in state space) of convergence for the approximate control is dictated by the stability region for the initial stabilizing control; it is therefore known *a priori* and is usually defined explicitly by the designer.
- The stability region of the approximate control is equal to the region of convergence.
- Finite truncations of the algorithm result in stable control laws that approximate the true optimal or robust solution arbitrarily closely.
- The computations are performed off-line. The on-line burden consists of computing linear combinations of state-dependent basis functions.
- Through judicious selection of the basis functions, the curse of dimensionality can be mitigated.
- Once the basis functions are selected, the approximation process can be completely automated.
- The resulting control is given by a linear combination of terms involving a set of basis functions. Essentially, this method is a series-based approximation.

Of course the method is not without its disadvantages. Most important, the curse of dimensionality is still an issue. As will be explained in § 3, the effect of the curse of dimensionality on the computational requirements can be effectively eliminated, but since the number of terms in an  $M$ th-order truncation of a complete basis grows

exponentially with the size of the state, if a standard complete basis (such as polynomial or sinusoidal) is blindly used in the algorithm (i.e. no basis selection by the designer), the amount of memory required to compute and store the coefficients grows exponentially. To overcome this problem, particular basis functions must be chosen which capture the essential dynamics of the problem. However, this creates some problems of its own. The selection of the basis functions requires some insight into the particular problem being solved. Further research needs to be directed at automating the selection of appropriate basis functions. Another disadvantage of the method is that tuning the control may not be computationally simple. However, if a quadratic penalty function is used in the optimal control case, then tuning the control becomes straightforward. An aspect of the algorithm that is not well understood is how the approximate solutions produced by the SGA algorithm relate to the viscosity solution for problems with discontinuous solutions to the HJI inequality.

The successive Galerkin approximation (SGA) algorithm for HJB equations first appeared in Beard (1995). An overview of the algorithm and some initial applications appear in Beard *et al.* (1996). The convergence of the Galerkin approximation of the generalized Hamilton–Jacobi–Bellman equation is shown in Beard *et al.* (1997), and the convergence of the SGA algorithm for the HJB equation is shown in Beard *et al.* (1998 a). The algorithm has also been applied to several problems. In McLain and Beard (1997) the SGA algorithm was used to synthesize nonlinear optimal controls for a hydraulically actuated positioning system. In McLain and Beard (1998 a) a nonlinear optimal control is designed for a missile autopilot. Finally, in McLain and Beard (1998 b) a nonlinear optimal control is designed for the planar motion of an underwater vehicle. The Galerkin method has also been used to synthesize a nonlinear filter that propagates an approximation of the entire conditional state density function. Results appear in Gunther *et al.* (1997), and Beard *et al.* (1998 b).

An independently developed, but closely related method has been reported in Georges (1995, 1996), where the Galerkin approximation method is applied directly to the HJB and HJI equations without the use of successive approximation.

This paper extends our previous work by developing a successive Galerkin approximation algorithm for the nonlinear  $\mathcal{H}_\infty$  control problem. We also prove the convergence of the successive approximation algorithm for the Hamilton–Jacobi–Isaacs equation. In addition, we show how the structure of the algorithm can be exploited to reduce the effect of the curse of dimensionality in the algorithm. This paper, in addition to our previous work represents a significant breakthrough in the control of nonlinear systems by enabling practical design of feedback control laws in the nonlinear  $\mathcal{H}_2$  and nonlinear  $\mathcal{H}_\infty$  settings.

In §2 the Successive Galerkin Approximation (SGA) algorithm for both the HJB and the HJI equations is derived. Since the focus of this paper is on approximating the HJB and HJI equations, we assume that the reader knows where these equations come from and how they are used in practical design. Brief convergence arguments that rely heavily on previously published results are given for the algorithms. Section 3 briefly explains how the exponential growth in the number of computations can be reduced to polynomial growth. In §4 we give four illustrative examples. The purpose of that section is not to analyse any particular example in detail but to show the breadth of applications to which these algorithms can be applied. Section 5 gives our conclusions.

## 2. Successive Galerkin Approximation

The objective of this section is to derive the Successive Galerkin Approximation (SGA) algorithms for the Hamilton–Jacobi–Bellman (HJB) and Hamilton–Jacobi–Isaacs (HJI) equations. The derivation is broken into two subsections. In §2.2 the HJB and HJI equations are reduced to an infinite sequence of linear partial differential equations termed Generalized Hamilton–Jacobi (GHJB and GHJI, respectively) equations. The successive approximation algorithm for the HJI equation has not previously appeared in the literature, and so a proof of its convergence is given. In §2.3, Galerkin’s method is used to approximate the Generalized Hamilton–Jacobi equations. Combining the two methods results in systematic approximation algorithms for the HJB and HJI equations which are easily implemented on a computer and which produce closed-loop stabilizing control laws on a well-defined region of state space.

### 2.1. Preliminaries

2.1.1. *Nonlinear  $\mathcal{H}_2$  control.* The nonlinear  $\mathcal{H}_2$  control problem is to minimize, with respect to the control, the integral

$$J(x_0) = \int_0^\infty \left( l(\varphi(t)) + \|u(\varphi(t))\|_R^2 \right) dt$$

where  $\varphi(t)$  is the solution of the state equation

$$\dot{x} = f(x) + g(x)u \tag{1}$$

$$x(0) = x_0 \tag{2}$$

and where  $x \in \mathbb{R}^n$  and  $u \in \mathbb{R}^m$  are the state and the control, respectively, and where  $\|u\|_R^2 \triangleq u^T R u$ . We assume throughout this paper that  $f(0) = 0$ . The state mapping  $f$  is assumed to be observable through the positive definite function  $l(\cdot)$ , and the matrix  $R$  is positive definite. The solution to this equation is well known to be given by the state feedback control

$$u^*(x) = -\frac{1}{2} R^{-1} g^T(x) \frac{\partial V^*}{\partial x}(x)$$

where  $V^*(x)$  satisfies the HJB equation (Lewis 1986)

$$\frac{\partial V^*}{\partial x} f + l - \frac{1}{4} \frac{\partial V^*}{\partial x} g R^{-1} g^T \frac{\partial V^*}{\partial x} = 0 \tag{3}$$

with the boundary condition  $V^*(0) = 0$ .

If system (1) is linear, and the state penalty function  $l$  is quadratic, then the HJB equation reduces to the standard Riccati equation of optimal control.

2.1.2. *Nonlinear  $\mathcal{H}_\infty$  control.* The nonlinear  $\mathcal{H}_\infty$  control problem is stated in terms of the  $\mathcal{L}_2$  gain of a system. Consider the system

$$\left. \begin{aligned} \dot{x} &= f(x) + g(x)u + k(x)w \\ y &= h(x) \end{aligned} \right\} \tag{4}$$

where  $x$  and  $u$  have the same dimension as above, and where  $w \in \mathbb{R}^q$  is the disturbance, and  $y \in \mathbb{R}^p$  is the output. We will assume throughout the paper that  $h(0) = 0$  and that  $f$  is observable through  $h$ .

System (4) is said to have  $\mathcal{L}_2$  gain less than or equal to  $\gamma$  if, for all  $T \geq 0$  and  $w \in \mathcal{L}_2(0, T)$ ,

$$\int_0^T (\|y(t)\|^2 + \|u(t)\|_R^2) dt \leq \gamma^2 \int_0^T \|w(t)\|_P^2 dt \quad (5)$$

where  $x(0) = 0$ . System (4) has  $\mathcal{L}_2$ -gain less than  $\gamma$  if there exists some  $0 \leq \tilde{\gamma} < \gamma$  such that equation (5) holds for  $\tilde{\gamma}$ .

The nonlinear  $\mathcal{H}_\infty$  (state-feedback) optimal control problem is to find, if it exists, the smallest value  $\gamma^* \geq 0$  and an associated state feedback control law  $u_\gamma^*(x)$  such that for any  $\gamma > \gamma^*$ , system (4) under the control of  $u_\gamma^*$  has  $\mathcal{L}_2$ -gain less than or equal to  $\gamma$ . As van der Schaft (1992) shows, a sufficient condition for system (4) to have  $\mathcal{L}_2$ -gain less than or equal to  $\gamma > 0$  is that

$$u_\gamma(x) = -\frac{1}{2} R^{-1} g^T(x) \frac{\partial V_\gamma}{\partial x}$$

where  $V_\gamma \geq 0$  is a smooth solution to the HJI equation

$$\frac{\partial V_\gamma^T}{\partial x} f + h^T h + \frac{1}{4} \frac{\partial V_\gamma^T}{\partial x} \left( \frac{1}{2\gamma^2} k P^{-1} k^T - g R^{-1} g^T \right) \frac{\partial V_\gamma}{\partial x} = 0 \quad (6)$$

with the boundary condition  $V_\gamma(0) = 0$ .

The main difference between the HJB and HJI equations is that the negative definite term  $-g R^{-1} g^T$  in the HJB equation is replaced by the (in general) indefinite term  $((1/2\gamma^2)k P^{-1} k^T - g R^{-1} g^T)$ . Another difference between the  $\mathcal{H}_2$  and  $\mathcal{H}_\infty$  control problems is the gain parameter  $\gamma$ . For  $\gamma$  greater than some lower bound  $\gamma_{LB}$  the HJI equation admits a continuously differentiable positive definite solution  $V_\gamma$  (Basar and Bernhard 1995); however, for  $\gamma < \gamma_{LB}$  there is no positive definite solution. We will exploit this fact at the end of §2.3, where we augment the SGA algorithm to find an approximation to the optimal nonlinear  $\mathcal{H}_\infty$  control law.

## 2.2. Successive approximation

Successive approximation, sometimes called 'iteration in policy space', was first used in the context of the HJB equation by Bellman (1957) to argue the existence of smooth solutions to the HJB equation. The use of successive approximation to approximate the HJB equation was placed on a rigorous mathematical foundation by Saridis and Lee (1979) and has also been studied in Vaisbord (1963), Rekasius (1964), Mil'shtein (1964), Leake and Liu (1967), Saridis and Balaram (1986) and Saridis and Wang (1994). In the context of the HJI equation, van der Schaft (1992) used similar arguments to argue the existence, under certain conditions, of smooth solutions to the HJI equation. However, van der Schaft only iterates on the control variable and does not reduce the HJI equation to a sequence of linear partial differential equations. To our knowledge this is the first time that iteration in policy space has been used to approximate the HJI equation.

2.2.1. *Nonlinear  $\mathcal{H}_2$  control.* The HJB equation (3) can be rewritten as

$$\frac{\partial V^{*T}}{\partial x}(f + gu^*) + l + \|u^*\|_R^2 = 0$$

$$u^* = -\frac{1}{2}R^{-1}g^T \frac{\partial V^*}{\partial x}$$

Writing the HJB equation in this form shows that the source of the nonlinearity in the HJB equation is the dependence of  $u^*$  on the value function. The basic idea of successive approximation is that instead of computing  $V^*$  and  $u^*$  simultaneously, we compute them iteratively, as shown below.

**Algorithm 1 (HJB: Successive Approximation):** Let  $u^{(0)}$  be an initial stabilizing control law for the system (1) with stability region  $\Omega$ .

For  $i = 0$  to  $\infty$   
 Solve for  $V^{(i)}$  from:

$$\frac{\partial V^{(i)T}}{\partial x}(f + gu^{(i)}) + l + \|u^{(i)}\|_R^2 = 0$$

Update the control:

$$u^{(i+1)} = -\frac{1}{2}R^{-1}g^T \frac{\partial V^{(i)}}{\partial x} \tag{7}$$

End

We can make the following statements about the sequence  $\{V^{(i)}\}_{i=0}^\infty$  and the sequence  $\{u^{(i)}\}_{i=0}^\infty$  (Saridis and Lee 1979, Saridis and Balaram 1986):

- For each  $x \in \Omega$ ,  $V^*(x) \leq V^{(i+1)}(x) \leq V^{(i)}(x)$ . Furthermore, if  $V^{(i+1)}(x) = V^{(i)}(x)$ , then  $V^{(i+1)}(x) = V^*(x)$ . This proves that  $V^{(i)} \rightarrow V^*$  pointwise on  $\Omega$ .
- If  $u^{(0)}$  stabilizes system (1) on  $\Omega$  then so does  $u^{(i)}$ , for all  $i \geq 0$ .
- The approximate control  $u^{(i)}$ ,  $i \geq 1$  is robust in the same sense as the optimal control  $u^*$  (i.e. 50% gain reduction margin and infinite gain margin) (Tsitsiklis and Athans 1984, Glad 1984, 1985, 1987).
- The control  $u^{(i+1)}$  has better performance than the control  $u^{(i)}$ .

The essence of Algorithm 1 is to reduce the HJB equation to an infinite sequence of linear partial differential equations of the form

$$\frac{\partial V^T}{\partial x}(f + gu) + l + \|u\|_R^2 = 0 \tag{8}$$

where  $u$  is a known function of  $x$ . This equation, termed the Generalized Hamilton–Jacobi–Bellman (GHJB) equation, is a linear first-order partial differential equation. Like the HJB equation, the GHJB is of fundamental importance in control theory. First, it is a Lyapunov equation that gives an explicit form for the control. In fact, the so-called Lyapunov optimal controls are derived from a GHJB equation (Robinett *et al.* 1997). In addition, the GHJB equation gives an explicit method to improve the ( $\mathcal{H}_2$ ) performance of any feedback control law. The difficulty is that the GHJB equation is difficult to solve analytically. In §2.3, Galerkin’s global approximation method will be used to construct an approximate solution to the GHJB equation.

2.2.2. Nonlinear  $\mathcal{H}_\infty$  control. The HJI equation (6) can be rewritten as



$$\frac{\partial V_\gamma^T}{\partial x} (f + gu_\gamma + kw_\gamma) + h^T h + \|u_\gamma\|_R^2 - \gamma^2 \|w_\gamma\|_P^2 = 0$$

$$u_\gamma = -\frac{1}{2} R^{-1} g^T \frac{\partial V_\gamma}{\partial x}$$

$$w_\gamma = \frac{1}{2} P^{-1} k^T \frac{\partial V_\gamma}{\partial x}$$

Writing the HJI equation in this form shows that the source of the nonlinearity in the HJI equation is the dependence of  $u_\gamma$  and  $w_\gamma$  on the value function  $V_\gamma$ . To reduce the HJI equation to a sequence of linear partial differential equations, we must use two simultaneous iterations of successive approximation. The algorithm is listed below.

**Algorithm 2** (HJI: successive approximation): *Let  $u^{(0)}$  be an initial stabilizing control law for the system (4) ( $w \equiv 0$ ) with stability region  $\Omega$ .*

*For  $i = 0$  to  $\infty$*

*Set  $w_\gamma^{(i,0)} \equiv 0$*

*For  $j = 0$  to  $\infty$*

*Solve for  $V_\gamma^{(i,j)}$  from:*

$$\frac{\partial V_\gamma^{(i,j)T}}{\partial x} (f + gu_\gamma^{(i)} + kw_\gamma^{(i,j)}) + h^T h + \|u_\gamma^{(i)}\|_R^2 - \gamma^2 \|w_\gamma^{(i,j)}\|_P^2 = 0 \quad (9)$$

*Update the Disturbance:*

$$w_\gamma^{(i,j+1)} = \frac{1}{2\gamma^2} P^{-1} k^T \frac{\partial V_\gamma^{(i,j)}}{\partial x} \quad (10)$$

*End*

*Update the Control:*

$$u_\gamma^{(i+1)} = -\frac{1}{2} R^{-1} g^T \frac{\partial V_\gamma^{(i,\infty)}}{\partial x} \quad (11)$$

*End*

This iteration has a game-theoretic interpretation. Given a fixed control  $u_\gamma^{(i)}$ , the inner loop updates the disturbance until it is the worst disturbance that the control  $u_\gamma^{(i)}$  can handle. After this worst-case disturbance is found, the control is updated to improve the performance of the system for that disturbance. The following two lemmas formalize these statements:

**Lemma 1:** *If for each  $(i,j)$  the system*

$$\dot{x} = f + gu_\gamma^{(i)} + kw_\gamma^{(i,j)} \quad (12)$$

*is asymptotically stable on  $\Omega$ , then*

$$V_\gamma^{(i,j)}(x) \leq V_\gamma^{(i,j+1)}(x) \leq V_\gamma^{(i,\infty)}(x)$$

*for each  $x \in \Omega$ , where  $V_\gamma^{(i,\infty)}$  solves the equation*

$$\frac{\partial V_\gamma^{(i,\infty)}}{\partial x} (f + gu_\gamma^{(i)}) + h^T h + \|u_\gamma^{(i)}\|_R^2 + \frac{1}{4\gamma^2} \frac{\partial V_\gamma^{(i,\infty)}}{\partial x} k P^{-1} k^T \frac{\partial V_\gamma^{(i,\infty)}}{\partial x} = 0 \quad (13)$$

Furthermore, if  $V_\gamma^{(i,j+1)}(x) = V_\gamma^{(i,j)}(x)$ , then  $V_\gamma^{(i,j+1)}(x) = V_\gamma^{(i,\infty)}$ , i.e.  $V_\gamma^{(i,j)} \rightarrow V_\gamma^{(i,\infty)}$  pointwise on  $\Omega$ .

**Proof:** Let  $x_0$  be any point in  $\Omega$ . Since system (12) is asymptotically stable for the pair  $(i, j + 1)$

$$V_\gamma^{(i,j)}(x_0) = - \int_0^\infty \dot{V}_\gamma^{(i,j)} dt$$

where the integration is taken along the solution of the system  $\dot{x} = f + gu_\gamma^{(i)} + kw_\gamma^{(i,j+1)}$  with initial condition  $x_0$ . Consequently

$$\begin{aligned} & V_\gamma^{(i,j)}(x_0) - V_\gamma^{(i,j+1)}(x_0) \\ &= \int_0^\infty \left( \frac{\partial V_\gamma^{(i,j+1)T}}{\partial x} (f + gu_\gamma^{(i)} + kw_\gamma^{(i,j+1)}) - \frac{\partial V_\gamma^{(i,j)T}}{\partial x} (f + gu_\gamma^{(i)} + kw_\gamma^{(i,j+1)}) \right) dt \\ &= \int_0^\infty \left( \gamma^2 \|w_\gamma^{(i,j+1)}\|_P^2 - \frac{\partial V_\gamma^{(i,j)}}{\partial x} kw_\gamma^{(i,j+1)} - \gamma^2 \|w_\gamma^{(i,j)}\|_P^2 + \frac{\partial V_\gamma^{(i,j)}}{\partial x} kw_\gamma^{(i,j)} \right) dt \\ &= -\gamma^2 \int_0^\infty \|w_\gamma^{(i,j)} - w_\gamma^{(i,j+1)}\|_P^2 dt \\ &\leq 0 \end{aligned}$$

where the first equality is obtained by the chain rule, the second equality is obtained by substituting equation (8) and simplifying, and the third equality is obtained by substituting equation (9) and simplifying. Therefore  $V_\gamma^{(i,j)}(x_0) \leq V_\gamma^{(i,j+1)}(x_0)$ . Identical arguments can be used to show that  $V_\gamma^{(i,j)}(x_0) \leq V_\gamma^{(i,\infty)}(x_0)$ , thus establishing the first claim. The second claim follows by substituting

$$w_\gamma^{(i,j+1)} = \frac{1}{2\gamma^2} P^{-1} k^T \frac{\partial V_\gamma^{(i,j+1)}}{\partial x}$$

into equation (9) and reducing to get an equation identical to equation (13) □

**Lemma 2:** If for each  $i$ , the system

$$\dot{x} = f + gu_\gamma^{(i)} + kw_\gamma^{(i,\infty)}$$

is asymptotically stable on  $\Omega$ , then

$$V_\gamma^{(i,\infty)}(x) \geq V_\gamma^{(i+1,\infty)}(x) \geq V_\gamma$$

for each  $x \in \Omega$ , where  $V_\gamma$  solves the HJI equation (6)

Furthermore, if  $V_\gamma^{(i+1,\infty)}(x) = V_\gamma^{(i,\infty)}(x)$ , then  $V_\gamma^{(i+1,\infty)}(x) = V_\gamma$ , i.e.  $V_\gamma^{(i,\infty)} \rightarrow V_\gamma$  pointwise on  $\Omega$ .

**Proof:** The proof of this lemma follows arguments that are identical to the proof of the preceding lemma. The proof is also essentially contained in van der Schaft (1992). □

The asymptotic stability of  $u_\gamma^{(i)}$  on  $\Omega$  is shown in the next lemma.

**Lemma 3:** If  $u_\gamma^{(i)}(x)$  is asymptotically stabilizing on  $\Omega$ , then so is  $u_\gamma^{(i+1)}(x)$ . Furthermore  $V_\gamma^{(i,\infty)}$  is a Lyapunov function for  $u_\gamma^{(i+1)}$ .

**Proof:** From equations (13) and (11)

$$\begin{aligned} \frac{\partial V_\gamma^{(i,\infty)T}}{\partial x}(f + gu_\gamma^{(i+1)}) &= -h^T h - \frac{1}{4\gamma^2} \frac{\partial V_\gamma^{(i,\infty)T}}{\partial x} k P^{-1} k^T \frac{\partial V_\gamma^{(i,\infty)}}{\partial x} - \|u_\gamma^{(i)}\|_R^2 \\ &\quad + \frac{\partial V_\gamma^{(i,\infty)T}}{\partial x} g u_\gamma^{(i+1)} - \frac{\partial V_\gamma^{(i,\infty)T}}{\partial x} g u_\gamma^{(i)} \\ &= -h^T h - \frac{1}{4\gamma^2} \frac{\partial V_\gamma^{(i,\infty)T}}{\partial x} k P^{-1} k^T \frac{\partial V_\gamma^{(i,\infty)}}{\partial x} - \|u_\gamma^{(i+1)}\|_R^2 \\ &\quad - \|u_\gamma^{(i)} - u_\gamma^{(i+1)}\|_R^2 \end{aligned}$$

which proves asymptotic stability, since the system dynamics are observable through  $h$ .  $\square$

As with Algorithm 1, the essence of Algorithm 2 is to reduce the HJI equation to an infinite sequence of linear partial differential equations of the form

$$\frac{\partial V^T}{\partial x}(f + gu + kw) + l + \|u\|_R^2 - \gamma^2 \|w\|_P^2 = 0 \quad (14)$$

where  $u$  and  $w$  are known functions of  $x$ . We term this equation the Generalized Hamilton–Jacobi–Isaacs (GHJI) equation. Like the GHJB equation, the GHJI equation is difficult to solve analytically. Section 2.3 uses Galerkin’s method to approximate this equation.

### 2.3. Galerkin approximation

For the reader’s convenience, we briefly summarize Galerkin’s method in the specific context of this paper. For a more rigorous and complete treatment, see Guenther and Lee (1988), Celia and Gray (1992) or Fletcher (1984). Given a partial differential equation  $\mathcal{A}(V) = 0$  with boundary conditions  $V(0) = 0$ , Galerkin’s method assumes that we can find a complete set of basis functions  $\{\phi_j\}_{j=1}^\infty$  such that  $\phi_j(0) = 0, \forall j$  and  $V(x) = \sum_{j=1}^\infty c_j \phi_j(x)$ , where the sum is assumed to converge pointwise in some set  $\Omega$ . An approximation to  $V$  is formed by truncating the series to  $V_N(x) = \sum_{j=1}^N c_j \phi_j(x)$ . The coefficients  $c_j$  are obtained by solving the algebraic equation

$$\int_{\Omega} \mathcal{A}(V_N(x)) \phi_\ell(x) dx = 0, \quad \ell = 1, \dots, N \quad (15)$$

In §2.3.1 we use Galerkin’s method to approximate the GHJB equation and show that, when this approximation is combined with Algorithm 1, the result is a practical algorithm for approximating nonlinear optimal control laws. Similarly, a practical algorithm for approximating nonlinear  $\mathcal{H}_\infty$  control laws is derived in §2.3.2.

**2.3.1. Nonlinear  $\mathcal{H}_2$  control.** Assume that  $u : \Omega \rightarrow \mathbb{R}^m$  is a feedback control law that asymptotically stabilizes system (1) on a compact set  $\Omega$ . Also assume that the set  $\{\phi_j\}_{j=1}^\infty$  is a complete basis set for the domain of the GHJB equation (8). Then, according to equation (15), an approximate solution to equation (8) is given by  $V_N = \sum_{j=1}^N c_j \phi_j$ , where the coefficients satisfy the equation

$$\int_{\Omega} \left[ \frac{\partial \left( \sum_{j=1}^N c_j \phi_j \right)^T}{\partial x} (f + gu) + l + \|u\|_R^2 \right] \phi_\ell dx = 0$$

$$\Rightarrow \int_{\Omega} \left[ \left( \sum_{j=1}^N c_j \frac{\partial \phi_j^T}{\partial x} \right) (f + gu) + l + \|u\|_R^2 \right] \phi_\ell dx = 0$$

Defining  $\mathbf{c}_N \triangleq (c_1, \dots, c_N)^T$ ,  $\Phi_N(x) \triangleq (\phi_1, \dots, \phi_N)^T$ , and  $\nabla \Phi_N \triangleq (\partial \phi_1 / \partial x, \dots, \partial \phi_N / \partial x)^T$  as the Jacobian of  $\Phi_N$ , we can then write these equations in vector form as

$$\int_{\Omega} \left[ \mathbf{c}_N^T \nabla \Phi_N (f + gu) + l + \|u\|_R^2 \right] \Phi_N dx = 0$$

$$\Rightarrow \mathbf{c}_N^T \left[ \int_{\Omega} \nabla \Phi_N (f + gu) \Phi_N dx \right] = - \int_{\Omega} (l + \|u\|_R^2) \Phi_N dx$$

$$\Rightarrow \left[ \int_{\Omega} \Phi_N (f + gu)^T \nabla \Phi_N^T dx \right] \mathbf{c}_N = - \int_{\Omega} (l + \|u\|_R^2) \Phi_N dx$$

$$\Rightarrow \left[ \left( \int_{\Omega} \Phi_N f^T \nabla \Phi_N^T dx \right) + \left( \int_{\Omega} \Phi_N u^T g^T \nabla \Phi_N^T dx \right) \right] \mathbf{c}_N$$

$$= - \int_{\Omega} l \Phi_N dx - \int_{\Omega} \|u\|_R^2 \Phi_N dx$$

$$\Rightarrow (\mathbf{A}_1^B + \mathbf{A}_2^B(u)) \mathbf{c}_N = \mathbf{b}_1^B + \mathbf{b}_2^B(u)$$

where  $\mathbf{A}_1^B$ ,  $\mathbf{A}_2^B(u)$ ,  $\mathbf{b}_1^B$  and  $\mathbf{b}_2^B(u)$  are defined appropriately, and where  $\mathbf{A}_1^B$  and  $\mathbf{b}_1^B$  are independent of the control  $u$ . Hence, given  $u$ , the GHJB equation (8) can be approximated by solving a number of  $n$  dimensional integrals and then a matrix inversion and multiplying.

At this point it appears that to use this approximation in Algorithm 1, we will need to recompute  $\mathbf{A}_2^B(u^{(i)})$  and  $\mathbf{b}_2^B(u^{(i)})$  at each step of the algorithm. Fortunately, this is not the case. To show why, use the approximate value function  $V_N$  to compute an updated control according to equation (7):

$$\hat{u}_N = -\frac{1}{2} R^{-1} g^T \frac{\partial V_N}{\partial x}$$

$$= -\frac{1}{2} R^{-1} g^T \sum_{j=1}^N c_j \frac{\partial \phi_j}{\partial x}$$

$$= -\frac{1}{2} R^{-1} g^T \nabla \Phi_N^T \mathbf{c}_N$$

To find the approximate value function for  $\hat{u}_N$ , we must compute  $\mathbf{A}_2^B(\hat{u}_N)$  and  $\mathbf{b}_2^B(\hat{u}_N)$ . But

$$\mathbf{A}_2^B(\hat{u}_N) = \int_{\Omega} \Phi_N \hat{u}_N^T g^T \nabla \Phi_N^T dx$$

$$= -\frac{1}{2} \int_{\Omega} \Phi_N \mathbf{c}_N^T \nabla \Phi_N g R^{-1} g^T \nabla \Phi_N^T dx$$

$$\begin{aligned}
&= -\frac{1}{2} \sum_{j=1}^N c_j \int_{\Omega} \Phi_N \frac{\partial \phi_j^T}{\partial x} g R^{-1} g^T \nabla \Phi_N^T dx \\
&= -\frac{1}{2} \sum_{j=1}^N c_j \mathbf{G}_j
\end{aligned}$$

where  $\mathbf{G}_j$  is defined appropriately. Similarly

$$\begin{aligned}
\mathbf{b}_2^B(\hat{u}_N) &= - \int_{\Omega} \Phi_N \hat{u}_N^T R^{-1} \hat{u}_N dx \\
&= -\frac{1}{4} \int_{\Omega} \Phi_N \mathbf{c}_N^T \nabla \Phi_N g R^{-1} g^T \nabla \Phi_N^T \mathbf{c}_N dx \\
&= -\frac{1}{4} \sum_{j=1}^N c_j \mathbf{G}_j \mathbf{c}_N
\end{aligned}$$

Therefore, the coefficients  $\mathbf{c}_N$  pull outside the integrals, and  $\mathbf{A}_2^B$  and  $\mathbf{b}_2^B$  can be computed iteratively once the matrices  $\{\mathbf{G}_j\}_{j=1}^N$  have been calculated.

The Galerkin approximation of the GHJB equation can be combined with Algorithm 1 to produce the algorithm below for computing an approximation to the HJI equation.

**Algorithm 3** (HJB: successive Galerkin approximation): *Let  $u^{(0)}$  be an initial stabilizing control law for the system (1) with stability region  $\Omega$ .*

*Pre-compute the integrals  $\mathbf{A}_1^B$ ,  $\mathbf{A}_2^B(u^{(0)})$ ,  $\mathbf{b}_1^B$ ,  $\mathbf{b}_2^B(u^{(0)})$ ,  $\{\mathbf{G}_j\}_{j=1}^{\infty}$*

*For  $i = 0$  to  $\infty$*

*Let*

$$\mathbf{A} = \begin{cases} \mathbf{A}_1^B + \mathbf{A}_2^B(u^{(0)}) & i = 0 \\ \mathbf{A}_1^B - \frac{1}{2} \sum_{j=1}^{\infty} c_j^{(i-1)} \mathbf{G}_j & i > 0 \end{cases}$$

$$\mathbf{b} = \begin{cases} \mathbf{b}_1^B + \mathbf{b}_2^B(u^{(0)}) & i = 0 \\ \mathbf{b}_1^B - \frac{1}{4} \sum_{j=1}^{\infty} c_j^{(i-1)} \mathbf{G}_j \mathbf{c}_{\gamma,N}^{(i-1)} & i > 0 \end{cases}$$

*Solve for  $V_N^{(i)} = \Phi_N^T \mathbf{c}_{\gamma,N}^{(i)}$  from:*

$$\mathbf{c}_{\gamma,N}^{(i)} = \mathbf{A}^{-1} \mathbf{b}$$

*Update the control:*

$$u_N^{(i+1)} = -\frac{1}{2} R^{-1} g^T \nabla \Phi_N^T \mathbf{c}_{\gamma,N}^{(i)} \quad (16)$$

*End*

There are several important points to notice about Algorithm 3.

- The control  $u_N^{(i)}$  is a feedback control law: Algorithm 3 essentially computes the coefficients in the linear combination of state-dependent basis functions that make up the control.
- The approximation algorithm can be computed off-line. Once the coefficients are computed, the control (16) can be calculated on-line in a number of different ways—using a digital computer or analogue hardware, for instance.
- It has been shown in Beard *et al.* (1998 a) that, for  $N$  sufficiently large, the control  $u_N^{(i)}$  is stabilizing on  $\Omega$  (for all  $i$ ) and is robust in the same sense as the optimal control  $u^*$ .
- It was also shown in Beard *et al.* (1998 a) that, as  $N \rightarrow \infty$ ,  $V_N^{(i)} \rightarrow V^*$  and  $u_N^{(i)} \rightarrow u^*$ .

2.3.2. *Nonlinear  $\mathcal{H}_\infty$  control.* As in the previous section, we assume that  $u_\gamma : \Omega \rightarrow \mathbb{R}^m$  is a feedback control law that asymptotically stabilizes system (4) ( $w \equiv 0$ ) on  $\Omega$ , and that the set  $\{\phi_j\}_{j=1}^\infty$  is a complete basis for the domain of the GHJI equation (14). Arguments identical to those in the previous section can be used to derive the following approximation to the GHJI equation:  $V_{\gamma,N} = \Phi_N \mathbf{c}_{\gamma,N}$ , where  $\mathbf{c}_{\gamma,N}$  satisfies the equation

$$(\mathbf{A}_1^I + \mathbf{A}_2^I(u) + \mathbf{A}_3^I(w))\mathbf{c}_{\gamma,N} = \mathbf{b}_1^I + \mathbf{b}_2^I(u) - \gamma^2 \mathbf{b}_3^I(w)$$

where

$$\begin{aligned} \mathbf{A}_1^I &= \int_{\Omega} \Phi_N f^T \nabla \Phi_N^T dx & \mathbf{A}_2^I(u_\gamma) &= \int_{\Omega} \Phi_N u_\gamma^T g^T \nabla \Phi_N^T dx \\ \mathbf{A}_3^I(w_\gamma) &= \int_{\Omega} \Phi_N w_\gamma^T k^T \nabla \Phi_N^T dx & \mathbf{b}_1^I &= - \int_{\Omega} \Phi_N h^T h dx \\ \mathbf{b}_2^I(u_\gamma) &= - \int_{\Omega} \Phi_N \|u_\gamma\|_R^2 dx & \mathbf{b}_3^I(w_\gamma) &= - \int_{\Omega} \Phi_N \|w_\gamma\|_P^2 dx \end{aligned}$$

Using equations (10) and (11) to update the disturbance and the control respectively, we obtain

$$\begin{aligned} \hat{w}_\gamma &= \frac{1}{2\gamma^2} P^{-1} k^T \nabla \Phi_N^T \mathbf{c}_{\gamma,N} \\ \hat{u}_\gamma &= -\frac{1}{2} R^{-1} g^T \nabla \Phi_N^T \mathbf{c}_{\gamma,N} \end{aligned}$$

As in the previous section, these expressions can be used to reduce  $\mathbf{A}_2^I(\hat{u}_\gamma)$ ,  $\mathbf{A}_3^I(\hat{w}_\gamma)$ ,  $\mathbf{b}_2^I(\hat{u}_\gamma)$ , and  $\mathbf{b}_3^I(\hat{w}_\gamma)$  as follows:

$$\begin{aligned} \mathbf{A}_2^I(\hat{u}_\gamma) &= -\frac{1}{2} \sum_{j=1}^N c_j \mathbf{G}_j & \mathbf{A}_3^I(\hat{w}_\gamma) &= \frac{1}{2\gamma^2} \sum_{j=1}^N c_j \mathbf{K}_j \\ \mathbf{b}_2^I(\hat{u}_\gamma) &= -\frac{1}{4} \sum_{j=1}^N c_j \mathbf{G}_j \mathbf{c}_{\gamma,N} & \mathbf{b}_3^I(\hat{w}_\gamma) &= -\frac{1}{4\gamma^4} \sum_{j=1}^N c_j \mathbf{K}_j \mathbf{c}_{\gamma,N} \end{aligned}$$

where

$$\mathbf{K}_j \triangleq \int_{\Omega} \Phi_N \frac{\partial \phi_j^T}{\partial x} k P^{-1} k^T \nabla \Phi_N^T dx$$

These equations can be used in Algorithm 2 to obtain the approximation algorithm below for the HJI equation.

**Algorithm 4** (HJI: successive Galerkin approximation): *Let  $u^{(0)}$  be an initial stabilizing control law for the system (4) ( $w \equiv 0$ ) with stability region  $\Omega$ .*

*Pre-compute the integrals  $A_1^I, A_2^I(u^{(0)}), b_1^I, b_2^I(u^{(0)}), \{G_j\}_{j=1}^\infty, \{K_j\}_{j=1}^\infty$ .*

*For  $i = 0$  to  $\infty$*

*Let*

$$\mathbf{A}^{(i)} = \begin{cases} \mathbf{A}_1^I + \mathbf{A}_2^I(u^{(0)}) & i = 0 \\ \mathbf{A}_1^I - \frac{1}{2} \sum_{k=1}^{\infty} c_k^{(i-1, \infty)} \mathbf{G}_k & i > 0 \end{cases}$$

$$\mathbf{b}^{(i)} = \begin{cases} \mathbf{b}_1^I + \mathbf{b}_2^I(u^{(0)}) & i = 0 \\ \mathbf{b}_1^I - \frac{1}{4} \sum_{k=1}^{\infty} c_k^{(i-1, \infty)} \mathbf{G}_k \mathbf{c}_{\gamma, N}^{(i-1, \infty)} & i > 0 \end{cases}$$

*For  $j = 0$  to  $\infty$*

$$\mathbf{A} = \begin{cases} \mathbf{A}^{(i)} & j = 0 \\ \mathbf{A}^{(i)} + \frac{1}{2\gamma^2} \sum_{k=1}^{\infty} c_k^{(ij-1)} \mathbf{G}_k & j > 0 \end{cases}$$

$$\mathbf{b} = \begin{cases} \mathbf{b}^{(i)} & j = 0 \\ \mathbf{b}^{(i)} + \frac{1}{4\gamma^2} \sum_{k=1}^{\infty} c_k^{(ij-1)} \mathbf{G}_k \mathbf{c}_{\gamma, N}^{(ij-1)} & j > 0 \end{cases}$$

*Solve for  $V_{\gamma, N}^{(i, j)}$  from:*

$$\mathbf{c}_{\gamma, N}^{(i, j)} = \mathbf{A}^{-1} \mathbf{b}$$

*Update the Disturbance:*

$$w_{\gamma, N}^{(ij+1)} = \frac{1}{2\gamma^2} P^{-1} k^T \nabla \Phi_N^T \mathbf{c}_{\gamma, N}^{(i, j)} \quad (17)$$

*End*

*Update the Control:*

$$u_\gamma^{(i+1)} = -\frac{1}{2} R^{-1} g^T \nabla \Phi_N^T \mathbf{c}_{\gamma, N}^{(i, \infty)} \quad (18)$$

*End*

The infinite iterations  $i = 0$  to  $\infty$  and  $j = 0$  to  $\infty$  cannot, of course, be implemented in the computer: reasonable stopping criteria must be implemented. The stopping criterion used in the inner loop is  $\|\mathbf{c}_{\gamma, N}^{(ij+1)} - \mathbf{c}_{\gamma, N}^{(ij)}\| < \epsilon$  for  $\epsilon$  small (typically 0.001). If this criterion is not met after 100 iterations, then the algorithm exits with a message that it has 'failed to converge'. When it does converge within 100 iterations,

the coefficients are labelled as  $\mathbf{c}_{\gamma,N}^{(i,\infty)}$ . The stopping criterion for the outer loop is  $\|\mathbf{c}_{\gamma,N}^{(i+1,\infty)} - \mathbf{c}_{\gamma,N}^{(i,\infty)}\| < \epsilon$ , with a similar interpretation about convergence.

The convergence of the Galerkin approximation to generalized Hamilton–Jacobi equations has been shown in Beard *et al.* (1997). The proof of the overall convergence of Algorithm 4 is beyond the scope of the present paper and will appear in a subsequent work.

To solve the nonlinear  $\mathcal{H}_\infty$  optimal control problem, we need to find the smallest  $\gamma$  for which the HJI equation has a positive definite solution. An approximation to this problem can be found by embedding Algorithm 4 in a bisection search algorithm for the optimal  $\gamma$ , where the test on whether  $\gamma$  is valid is whether or not Algorithm 4 converges.

### 3. Mitigating the curse of dimensionality

The curse of dimensionality arises in two ways in Algorithms 3 and 4. The first source is the number of multidimensional integrals that need to be calculated. If they are computed using numerical quadrature, the number of function calls needed during the computation of each integral will increase exponentially with the dimension of the state space  $n$ . The second source of the curse of dimensionality is that the number of basis elements needed to form a complete basis of order  $M$  (e.g. fourth order) grows exponentially with the dimension of the state space. As the number of basis elements grows, an increasing amount of memory is needed to store the coefficients in the approximation, causing run-time execution problems.

We will demonstrate the problem and our solution with a concrete example. Suppose that the dimension of the state space is  $n = 2$  and the desired order of approximation is  $M = 3$ . The second order truncation of a complete set of polynomial basis functions is given by

$$\Phi_9 = (1, x_1, x_1^2, x_2, x_1x_2, x_1^2x_2, x_2^2, x_1x_2^2, x_1^2x_2^2)^T$$

Now suppose that we would like to compute the vector  $\mathbf{b}_1^B = \int_\Omega \Phi_N(x)l(x) dx$ . To do so, we need to calculate the nine two-dimensional integrals

$$\begin{aligned} &\int_\Omega l(x) dx, & \int_\Omega l(x)x_1 dx, & \int_\Omega l(x)x_1^2 dx \\ &\int_\Omega l(x)x_2 dx, & \int_\Omega l(x)x_1x_2 dx, & \int_\Omega l(x)x_1^2x_2 dx \\ &\int_\Omega l(x)x_2^2 dx, & \int_\Omega l(x)x_1x_2^2 dx, & \int_\Omega l(x)x_1^2x_2^2 dx \end{aligned}$$

By making two key assumptions, we can dramatically reduce the computational load. The first assumption is that  $l(x)$  is a separable function. In other words,  $l(x)$  has the structure:

$$l(x) = \sum_{p=1}^P l_{1p}(x_1)l_{2p}(x_2).$$

This structure is very general and includes, for example, quadratic penalty functions, polynomial functions and sinusoidal functions. The second assumption is that  $\Omega$  is



rectangular (i.e.  $\Omega = [a_1, b_1] \times [a_2, b_2]$ ). The user must therefore find a rectangular subset of the stability region of the initial stabilizing control  $u^{(0)}$ .

To illustrate the basic idea, suppose that  $l(x) = l_1(x_1)l_2(x_2)$  and that  $\Omega = [a_1, b_1] \times [a_2, b_2]$ ; then the multidimensional integrals separate as follows:

$$\begin{aligned} & \int_{a_1}^{b_1} l_1 dx_1 \int_{a_2}^{b_2} l_2 dx_2, & \int_{a_1}^{b_1} l_1 x_1 dx_1 \int_{a_2}^{b_2} l_2 dx_2, & \int_{a_1}^{b_1} l_1 x_1^2 dx_1 \int_{a_2}^{b_2} l_2 dx_2 \\ & \int_{a_1}^{b_1} l_1 dx_1 \int_{a_2}^{b_2} l_2 x_2 dx_2, & \int_{a_1}^{b_1} l_1 x_1 dx_1 \int_{a_2}^{b_2} l_2 x_2 dx_2, & \int_{a_1}^{b_1} l_1 x_1^2 dx_1 \int_{a_2}^{b_2} l_2 x_2 dx_2 \\ & \int_{a_1}^{b_1} l_1 dx_1 \int_{a_2}^{b_2} l_2 x_2^2 dx_2, & \int_{a_1}^{b_1} l_1 x_1 dx_1 \int_{a_2}^{b_2} l_2 x_2^2 dx_2, & \int_{a_1}^{b_1} l_1 x_1^2 dx_1 \int_{a_2}^{b_2} l_2 x_2^2 dx_2 \end{aligned}$$

Note that, with this partition, we must compute a total of  $M^n = 3^2$   $n$ -dimensional quadratures, or if  $l$  is separable and  $\Omega$  is rectangular, we must compute  $nM^n = 2 \cdot 3^2$  one-dimensional quadratures. However, the above expressions show that there is a tremendous amount of redundancy in the integrals being computed. By removing this redundancy we will reduce the number of one-dimensional quadratures to  $nM$  (i.e. linear growth in  $n$ ).

Using tensor product notation, we can write

$$\Phi_9 = \begin{pmatrix} 1 \\ x_1 \\ x_1^2 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ x_2 \\ x_2^2 \end{pmatrix}$$

and

$$\mathbf{b}_1^B = - \left( \int_{a_1}^{b_1} \begin{pmatrix} 1 \\ x_1 \\ x_1^2 \end{pmatrix} l_1 dx_1 \right) \otimes \left( \int_{a_2}^{b_2} \begin{pmatrix} 1 \\ x_2 \\ x_2^2 \end{pmatrix} l_2 dx_2 \right)$$

By performing the quadratures first and the tensor product second, we reduce the number of one-dimensional quadratures to  $nM$ . In general, if  $\varphi_M(x)$  is an  $M$ th order truncation of a complete basis on  $\mathbb{R}$ , and if  $\Phi_N = \varphi_M(x_1) \otimes \cdots \otimes \varphi_M(x_n)$  and  $l(x) = \sum_{j=1}^Q l_{1j}(x_1) \cdots l_{nj}(x_n)$  and  $\Omega = [a_1, b_1] \times \cdots \times [a_n, b_n]$  then

$$\mathbf{b}_1^B = - \sum_{j=1}^Q \left( \int_{a_1}^{b_1} l_{1j}(x_1) \varphi_M(x_1) dx_1 \right) \otimes \cdots \otimes \left( \int_{a_n}^{b_n} l_{nj}(x_n) \varphi_M(x_n) dx_n \right)$$

Similar expressions can be derived for  $\mathbf{A}_1^B$ ,  $\mathbf{A}_2^B(u^{(0)})$ ,  $\mathbf{b}_2^B(u^{(0)})$ ,  $\mathbf{G}_j$ ,  $\mathbf{A}_1^I$ ,  $\mathbf{A}_2^I(u^{(0)})$ ,  $\mathbf{b}_1^I$ ,  $\mathbf{b}_2^I(u^{(0)})$ , and  $\mathbf{K}_j$ ; these are developed explicitly in Lawton and Beard (1998).

At the expense of requiring that all system equations are sums of separable functions and that  $\Omega$  is rectangular, the approach solves the problem of exponential growth in the number of computations required to implement the algorithm. It does not, however, reduce the number of basis functions required for a complete basis. This problem, fundamental to approximation theory, is typically resolved through a careful, albeit heuristic, selection of approximating functions. We adopt this approach in §4.

4. Illustrative examples

The purpose of this section is to demonstrate the usefulness of the SGA approach for designing controllers. To do this, four different example applications are summarized below. Each example represents a real physical system of practical interest and has significant nonlinearities in its equations of motion. Two nonlinear  $\mathcal{H}_2$  optimal control examples are considered first, followed by two nonlinear  $\mathcal{H}_\infty$  control examples.

4.1. Nonlinear  $\mathcal{H}_2$  control

4.1.1. *Hydraulic actuation system.* The first nonlinear  $\mathcal{H}_2$  optimal control example considered is the electro-hydraulic actuation system shown in figure 1. A more detailed discussion of this system and its control implementation can be found in McLain and Beard (1997). The dynamics of this system are described by the equations of motion

$$\begin{aligned} \dot{P}_L &= \frac{4B}{V} \left[ C_d h x_v \sqrt{\frac{P_s - P_L \text{sgn}(x_v)}{\rho}} - A v_p \right] \\ \dot{v}_p &= \frac{1}{m_L} [A P_L - b v_p - F_c \text{sgn}(v_p)] \\ \dot{x}_p &= v_p \\ \dot{x}_v &= -\frac{1}{\tau} x_v + \frac{K_s}{\tau} i_v \end{aligned}$$

where  $P_L$  is the pressure difference across the piston,  $v_p$  is the piston velocity,  $x_p$  is the piston position,  $x_v$  is the valve position, and  $i_v$  is the valve current. The functions  $f(x)$  and  $g(x)$  can be easily identified by placing the equations of motion in the form of equation (1).

The domain of possible values for the states,  $\Omega$ , was determined from the physical constraints on the system

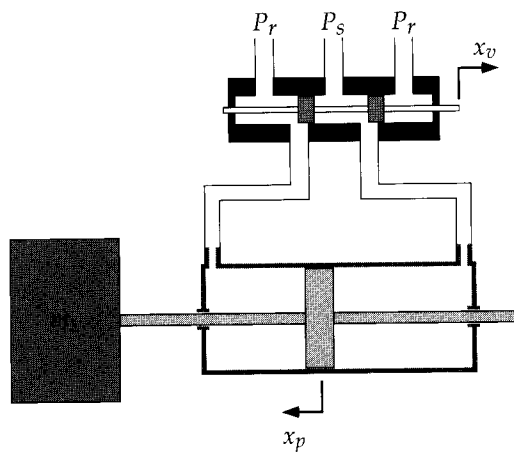


Figure 1. Hydraulic system schematic.

$$\begin{aligned}
 -P_s &\leq P_L \leq P_s \\
 -Q_{\max}/A &\leq v_p \leq Q_{\max}/A \\
 x_{p_{\min}} &\leq x_p \leq x_{p_{\max}} \\
 x_{v_{\min}} &\leq x_v \leq x_{v_{\max}}
 \end{aligned}$$

where  $Q_{\max}$  is the maximum rated no-load flow through the valve,  $x_{p_{\min}}$  and  $x_{p_{\max}}$  are bounds on the displacement of the piston, and  $x_{p_{\min}}$  and  $x_{v_{\max}}$  are bounds on the displacement of the valve spool.

For this problem, the basis functions were chosen to be

$$\{\phi_j\}_{j=1}^{10} = \{P_L^2, v_p^2, x_p^2, x_v^2, P_L v_p, P_L x_p, P_L x_v, v_p x_p, v_p x_v, x_p x_v\}$$

Other basis functions were evaluated (for example,  $\sqrt{P_s - P_L}$ ) but failed to contribute significantly to the control and therefore were not included in the results presented.

In this problem, regulation to a non-zero piston-position state was considered. This is easily handled by replacing the  $x_p$  state with the error state  $e = r - x_p$ , where  $r$  is the desired piston position.

Controllers for this system were designed using three different performance indices, which are tabulated along with their corresponding control laws in table 1. Simulation results are presented in figure 2. For each controller the initial piston position was  $-5$  cm and the piston was given a step command to move to the  $5$  cm position. The initial response of the system was obtained with cost on the states set to  $l = P_L^2 + v_p^2 + e^2$  and the cost on the control set to  $R = 1$ . With these costs the response of the system is slower than desired.

To improve the speed of response, the cost on the state errors was changed to  $l = 20P_L^2 + 0.2v_p^2 + 10e^2$ . As expected, the resulting response is much quicker, while the amount of control used (as indicated by the valve position motion) is significantly larger. The amount of control used can be moderated by increasing  $R$ . This is demonstrated in figure 2, where the cost  $l = 20P_L^2 + 0.2v_p^2 + 10e^2$  was coupled with  $R = 200$ . It can be seen that less control is used and the response is slightly slower.

---

Initial design:

$$\begin{aligned}
 l &= P_L^2 + v_p^2 + (r - x_p)^2 & R &= 1 \\
 u^{(7)} &= -0.303P_L - 0.570v_p + 1.31(r - x_p) - 724x_v \\
 V^{(0)} &= 376 \Rightarrow V^{(7)} = 113
 \end{aligned}$$

High-performance design:

$$\begin{aligned}
 l &= 20P_L^2 + 0.2v_p^2 + 10(r - x_p)^2 & R &= 1 \\
 u^{(9)} &= -2.51P_L + 0.373v_p + 4.41(r - x_p) - 2340x_v \\
 V^{(0)} &= 658 \Rightarrow V^{(9)} = 392
 \end{aligned}$$

Moderate control design:

$$\begin{aligned}
 l &= 20P_L^2 + 0.2v_p^2 + 10(r - x_p)^2 & R &= 200 \\
 u^{(5)} &= -0.0320P_L + 0.0470v_p + 0.349(r - x_p) - 146x_v \\
 V^{(0)} &= 1870 \Rightarrow V^{(5)} = 712
 \end{aligned}$$


---

Table 1. Optimal controller designs, hydraulic system.

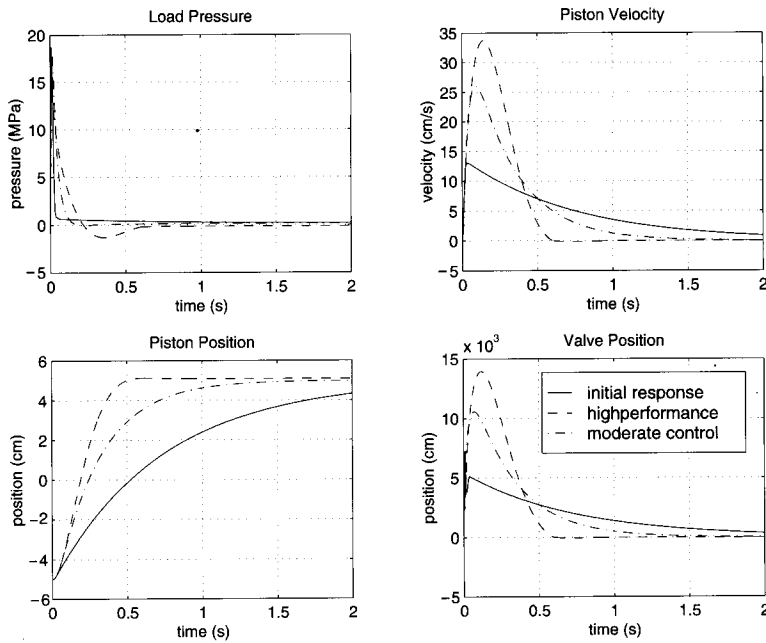


Figure 2. Optimal control responses—hydraulic system.

4.1.2. *Missile autopilot design.* In this example the design of a pitch-axis autopilot for a tail-controlled missile is considered. A schematic diagram of such a system is shown in figure 3. The objective of the autopilot is to control the  $z$ -axis acceleration of the missile using tail-fin deflection commands. An in-depth discussion of this system and its control design can be found in McLain and Beard (1998 a).

The dynamic model for this system was taken from Reichert (1990) and is represented by the following equations of motion:

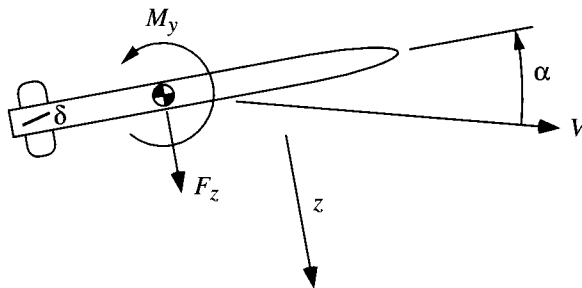


Figure 3. Missile system—pitch axis.

$$\begin{aligned}\dot{q} &= \frac{M_y}{I_y} \\ \dot{\alpha} &= \frac{\cos^2(\alpha)}{mU} F_z + q \\ \ddot{\delta} &= -2\zeta\omega_n\dot{\delta} - \omega_n^2\delta + \omega_n^2\delta_c\end{aligned}$$

where  $q$  is the pitch rate,  $\alpha$  is the angle of attack,  $\delta$  is the tail fin deflection, and  $\delta_c$  is the commanded tail fin deflection. The pitch moment,  $M_y$ , and the lift,  $F_z$ , are given by

$$\begin{aligned}M_y &= C_m Q S d \\ F_z &= C_n Q S\end{aligned}$$

where  $Q$  is dynamic pressure,  $S$  is a reference area,  $d$  is the missile diameter, and  $C_m$  and  $C_n$  are aerodynamic coefficients given by

$$\begin{aligned}C_n &= a_1\alpha^3 + a_2\alpha|\alpha| + a_3\alpha + a_4\delta \\ C_m &= b_1\alpha^3 + b_2\alpha|\alpha| + b_3\alpha + b_4\delta\end{aligned}$$

The output of interest,  $z$ -axis acceleration (in units of  $g$ ), is calculated by

$$\eta_z = \frac{F_z}{mg}$$

The control objective is to regulate  $\eta_z$  to a non-zero steady-state value. To do so, the states  $q$ ,  $\alpha$  and  $\delta$ , and the control  $\delta_c$ , must go to non-zero steady-state values. The underlying theory of the SGA algorithm requires that  $V(x)$  be finite, and therefore that the states and control go to zero as time goes to infinity. To apply Algorithm 3, a change of variables must be performed to satisfy this requirement.

For a given desired steady-state acceleration  $\eta_{z,d}$ , corresponding steady-state values for the states and the control can be calculated numerically from the equations of motion and the output equation. By performing the change of variables

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} q - q_{ss} \\ \alpha - \alpha_{ss} \\ \dot{\delta} - \dot{\delta}_{ss} \\ \delta - \delta_{ss} \end{bmatrix}$$

for the states and  $u = \delta_c - \delta_{c,ss}$  for the control, a new set of state equations results, where a stabilizing control will cause  $\eta_z \rightarrow \eta_{z,d}$ ,  $x \rightarrow 0$ , and  $u \rightarrow 0$  as  $t \rightarrow \infty$ , thereby enabling the application of Algorithm 3.

For this problem and the results presented here, the domain of possible values for the states was defined to be

$$\begin{aligned}-200 \text{ deg/s} &\leq x_1 \leq 200 \text{ deg/s} \\ -25 \text{ deg} &\leq x_2 \leq 25 \text{ deg} \\ -800 \text{ deg/s} &\leq x_3 \leq 800 \text{ deg/s} \\ -25 \text{ deg} &\leq x_4 \leq 25 \text{ deg}\end{aligned}$$

Initial Design, 15g

$$l = (\eta_{z_d} - \eta_z)^2 \quad R = 1$$

$$u^{(4)} = 0.1342(q - q_{ss}) + 0.1319(\alpha - \alpha_{ss}) - 0.01413(\dot{\delta} - \dot{\delta}_{ss}) - 1.143(\delta - \delta_{ss}) - 1.52 \times 10^{-7}(\alpha - \alpha_{ss})|\alpha - \alpha_{ss}| + 3.20 \times 10^{-7}(\alpha - \alpha_{ss})^3$$

$$V^{(0)} = 69.9 \Rightarrow V^{(4)} = 42.8$$

High-performance design, 15g

$$l = (\eta_{z_d} - \eta_z)^2 + 0.5x_1^2 + 25x_2^2 \quad R = 1$$

$$u^{(8)} = 0.07901(q - q_{ss}) + 1.041(\alpha - \alpha_{ss}) - 0.04360(\dot{\delta} - \dot{\delta}_{ss}) - 4.813(\delta - \delta_{ss}) - 1.18 \times 10^{-6}(\alpha - \alpha_{ss})|\alpha - \alpha_{ss}| + 2.52 \times 10^{-6}(\alpha - \alpha_{ss})^3$$

$$V^{(0)} = 1480 \Rightarrow V^{(8)} = 635$$

Table 2. Optimal missile autopilot designs.

A standard second-order set of basis functions ( $\phi_1$  through  $\phi_{10}$ ) was augmented with several other basis functions:

$$\{\phi_j\}_{j=1}^{12} = \{x_1^2, x_1x_2, x_2^2, x_1x_3, x_2x_3, x_3^2, x_1x_4, x_2x_4, x_3x_4, x_4^2, x_2|x_2|x_3, x_2^3x_3\}$$

These additional basis functions were selected in an attempt to better capture the dynamics of the system in approximating the cost function  $V^{(i)}$  and to introduce nonlinear terms into the resulting feedback control law.

Controllers were designed for a 15g acceleration command using two different cost functions  $l(x)$ . For the results presented, the cost on the control,  $R$ , was set to one. The resulting controller designs can be found in table 2. Figure 4 shows the responses obtained for each of the two controllers.

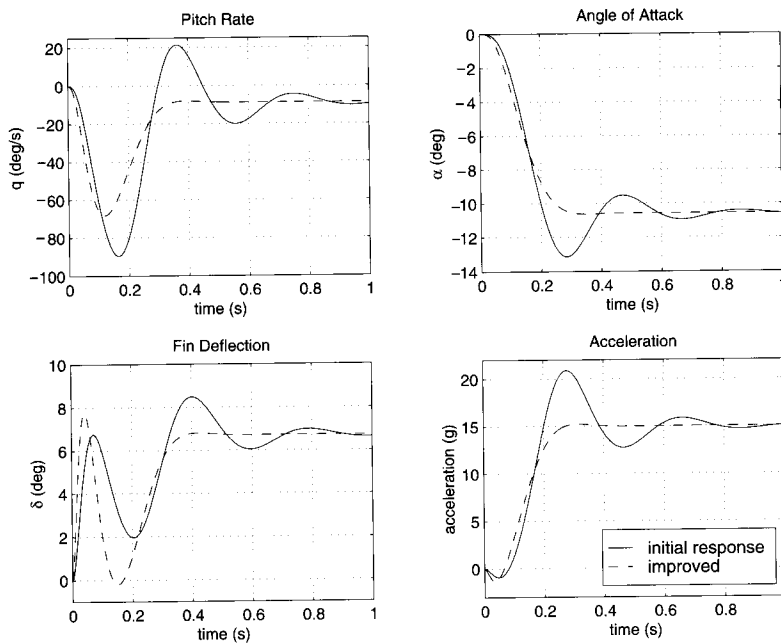


Figure 4. Missile system response to 15g acceleration command.

The first cost function was chosen to be the square of the difference between the desired and actual accelerations:  $l = (\eta_{zd} - \eta_z)^2$ . The resulting response of the system was satisfactory but exhibited significant overshoot and a long settling time. To improve the performance of the system a new cost function was formulated. Damping was added to the system by including a cost on pitch-rate error, while the speed of response was maintained by including a cost on angle-of-attack error in the new cost function:  $l = (\eta_{zd} - \eta_z)^2 + 0.5x_1^2 + 25x_2^2$ . As can be seen in figure 4, the performance of the missile is significantly improved with the new weighting function.

#### 4.2. Nonlinear $\mathcal{H}_\infty$ control

4.2.1. *Underwater vehicle system.* In this example, Algorithm 4 is used to synthesize a nonlinear  $\mathcal{H}_\infty$  controller to control the forward motion of an underwater vehicle in the presence of significant disturbances and having significant model uncertainty. The equations of motion which describe the forward motion of the vehicle are given by

$$\begin{aligned}\dot{u} &= \frac{1}{m + m_A}(-bu|u| + T + d) \\ \dot{x} &= u \\ \dot{T} &= -\frac{1}{\tau}T + \frac{k_T}{\tau}i\end{aligned}$$

where  $u$  is the forward velocity of the vehicle,  $x$  is the position of the vehicle,  $m$  is the mass (in air) of the vehicle,  $m_A$  is the added mass of the water surrounding the vehicle,  $b$  is the square-law drag coefficient,  $T$  is the applied thrust,  $d$  is an external disturbance caused by currents or forces from an attached tether,  $\tau$  is the thruster time constant,  $k_T$  the thrust coefficient, and  $i$  is the current command to the thruster. Typically, values for  $m_A$  and  $b$  are not known accurately because of the difficulty in measuring them and because they are dependent on the operating conditions of the vehicle. In this study the value for  $b$  is varied up to  $\pm 90\%$  of its nominal value. Letting  $\hat{b}$  represent the nominal value of  $b$  and letting  $\delta b = b - \hat{b}$ , the equations of motion can be written

$$\begin{aligned}\dot{u} &= \frac{1}{m + m_A}(-bu|u| + T + d - \delta b u|u|) \\ \dot{x} &= u \\ \dot{T} &= -\frac{1}{\tau}T + \frac{k_T}{\tau}i\end{aligned}$$

Putting the system into the standard form of equation (4) results in

$$\begin{pmatrix} \dot{u} \\ \dot{x} \\ \dot{T} \end{pmatrix} = \begin{pmatrix} -\frac{b}{m + m_A}u|u| \\ u \\ -\frac{1}{\tau}T \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ \frac{k_T}{\tau} \end{pmatrix} i + \begin{pmatrix} -\frac{1}{m + m_A}u|u| & \frac{1}{m + m_A} \\ 0 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \delta b \\ d \end{pmatrix}$$

Though the output of interest for this system is the forward displacement  $x$ , for the purpose of obtaining satisfactory transient response the output was chosen to be  $10x_1 + 15x_2$  for the design of the control. Values for the model parameters used to design the control and simulate the performance of the closed-loop system are based on those of the NEROV vehicle (Fossen and Sagatun 1991).

For the design of the controller, the basis functions were chosen to be

$$\{\phi_j\}_{j=1}^8 = \{u^2, x^2, T^2, ux, uT, xT, T^4, x^3T\}$$

For the simulation results presented, station-keeping controllers were designed to regulate the velocity and position of the vehicle to zero. Using Algorithm 4, the following robust control law was obtained:

$$i = -11.9u - 25.6x - 0.490T - 0.0050T^3 - 5.68x^3$$

The optimal value of  $\gamma$  corresponding to this control law was found to be 0.180. Figure 5 shows the response of the underwater vehicle system to initial conditions (0.5 m/s on the velocity state and 1.0 m on the position state) for the system with the maximum and minimum perturbations in the drag coefficient,  $\delta b_{\max}$  and  $\delta b_{\min}$ . Though  $b$  is varied significantly, the similarity of results demonstrates the robustness of the controller.

In the ocean environment, disturbances are commonplace. Figure 6 shows a disturbance time history that is typical of what a tether might apply under actual operating conditions, as well as the response of the system to this disturbance. As in figure 5, the states were given initial conditions of 0.5 m/s and 1.0 m. These results demonstrate the ability of the controller to reject significant disturbances.

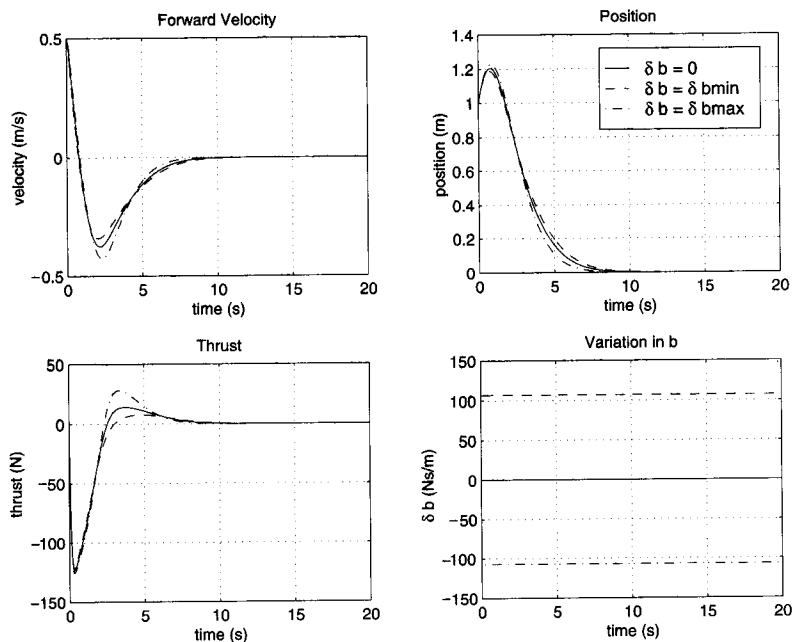


Figure 5. State trajectories with maximum parametric uncertainty.



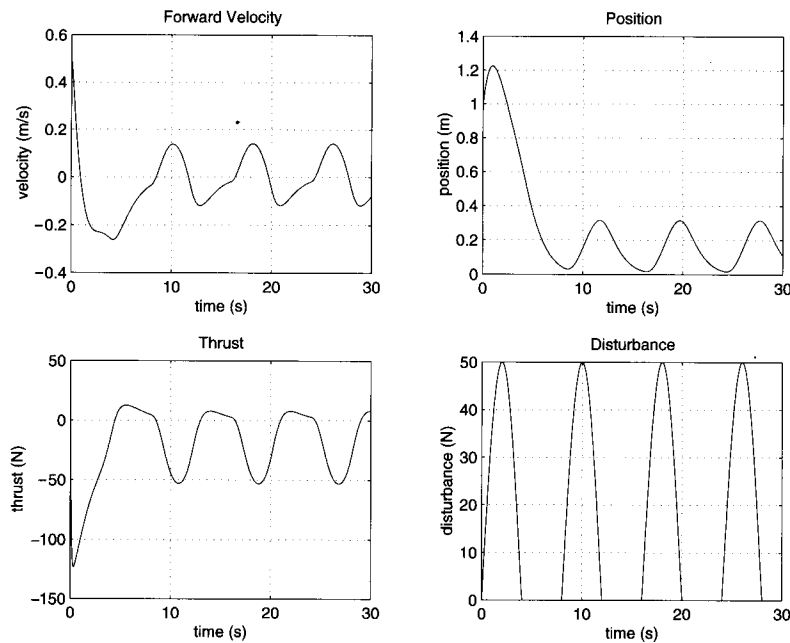


Figure 6. Tether force disturbance rejection.

4.2.2. *Hydraulic actuation system.* As a second example of nonlinear  $\mathcal{H}_\infty$  controller design using Algorithm 4, the hydraulic actuation system of figure 1 is reconsidered. The objective here is to design a control law that is robust to significant variations in the fluid parameters of the hydraulic system, namely the bulk modulus  $B$  and the density  $\rho$ . In practice, these parameters are often uncertain because of their sensitivity to changes in fluid temperature and because of air becoming entrained in the hydraulic fluid.

If the nominal values for bulk modulus and density are represented as  $\hat{B}$  and  $\hat{\rho}$ , respectively, then the load pressure equation of motion for the hydraulic system can be rewritten as

$$\dot{P}_L = \frac{4\hat{B}}{V} \left( C_d h x_v \frac{\sqrt{P_s - P_L \text{sgn}(x_v)}}{\hat{\rho} - A v_p} \right) + \frac{4\hat{B}}{V} \left( C_d h x_v \frac{\sqrt{P_s - P_L \text{sgn}(x_v)}}{\hat{\rho}} \right) \mu - \frac{4\hat{B} A v_p}{V} \beta$$

where  $\mu$  is the uncertainty in the quotient  $B/\sqrt{\rho}$  and  $\beta$  is the uncertainty in  $B$ :

$$\frac{B}{\sqrt{\rho}} = (1 + \mu) \frac{\hat{B}}{\sqrt{\hat{\rho}}}$$

$$B = (1 + \beta) \hat{B}$$

For the design of the controller, the basis functions used were the same as those used for the nonlinear  $\mathcal{H}_2$  design. To ensure that different states were weighted appropriately in the synthesis of the control, the output  $y$  was selected to be

$$y = P_L + v_p + x_p$$

Application of Algorithm 4 resulted in the  $\mathcal{H}_\infty$  control law

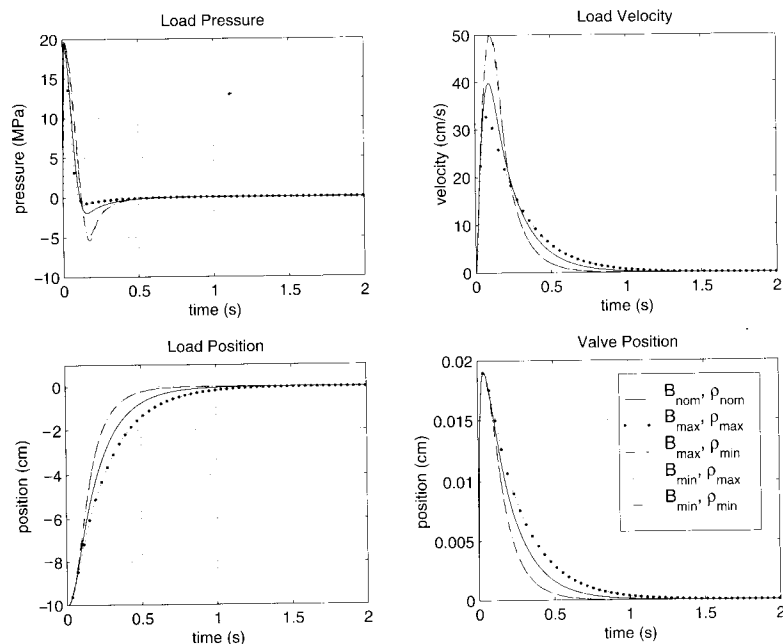


Figure 7. Hydraulic system response with maximum parametric uncertainty.

$$\dot{i}_v = 0.0112P_L + 0.297v_p - 11.9x_p - 31.8x_v$$

and an optimal value of  $\gamma = 0.070$ . Figure 7 shows the results obtained using this control law. The bulk modulus was allowed to vary  $\pm 50\%$  from its nominal value, while the fluid density was allowed to vary  $\pm 30\%$  from its nominal value. Such variations, though extreme, could occur if air became entrained in the hydraulic fluid. In figure 7, state trajectories are plotted for the following perturbation cases:  $[\hat{B}, \hat{\rho}]$ ,  $[B_{\max}, \rho_{\max}]$ ,  $[B_{\max}, \rho_{\min}]$ ,  $[B_{\min}, \rho_{\max}]$ , and  $[B_{\min}, \rho_{\min}]$ . The results demonstrate both the ability to design nonlinear  $\mathcal{H}_\infty$  controllers and the robustness of such a controller to significant perturbations in the parameters of the system.

## 5. Conclusions

Algorithms that approximate the solution of the Hamilton–Jacobi–Bellman and Hamilton–Jacobi–Isaacs equations have been derived using successive approximation and Galerkin’s method. The algorithms, performed off-line, produce explicit expressions for feedback control laws which have a well-defined region of stability. The structure of the algorithms was exploited to reduce the computational growth from exponential in the dimension of the state space to polynomial. The algorithms have been applied to four physical examples to illustrate the generality of the results. Since these algorithms allow efficient approximations to the nonlinear (state-feedback)  $\mathcal{H}_2$  optimal control and nonlinear (state-feedback)  $\mathcal{H}_\infty$  control problems, we believe that this paper represents a significant breakthrough in the control of nonlinear systems.

## References

- ATHANS, M., and FALB, P., 1966, *Optical Control* (New York: McGraw-Hill).
- BALL, J. A., HELTON, J. W., and WALKER, M. L., 1993,  $H_\infty$  control for nonlinear systems with output feedback. *IEEE Transactions on Automatic Control*, **38**, 548–559.
- BASAR, T., and BERNHARD, P., 1995,  *$H_\infty$ -Optimal Control and Related Minimax Design Problems* (Birkhäuser).
- BEARD, R., 1995, Improving the closed-looped performance of nonlinear systems. PhD thesis, Rensselaer Polytechnic Institute, Troy, New York.
- BEARD, R., SARIDIS, G., and WEN, J., 1996, Improving the performance of stabilizing control for nonlinear systems. *Control Systems Magazine*, **16**, 27–35.
- BEARD, R., SARIDIS, G., and WEN, J., 1997, Galerkin approximation of the generalized Hamilton–Jacobi–Bellman equation. *Automatica* **33**, 2159–2177.
- BEARD, R., SARIDIS, G., and WEN, J., 1998a, Approximate solutions to the time-invariant Hamilton–Jacobi–Bellman equation. *Journal of Optimization Theory and Applications*.
- BEARD, R. W., GUNTHER, J., LAWTON, J., and STIRLING, W., 1998b, The nonlinear projection filter. *AIAA Journal of Guidance, Control and Dynamics*, **96**, 589–626.
- BELLMAN, R. E., 1957, *Dynamic Programming* (Princeton, NJ: Princeton University Press).
- BITTANTI, S., LAUB, A., and WILLEMS, J. C., 1991, *The Riccati Equation* (New York: Springer).
- CELIA, M. A., and GRAY, W. G., 1992, *Numerical Methods for Differential Equations* (Englewood Cliffs, NJ: Prentice-Hall).
- CRANDALL, M. G., ISHII, H., and LIONS, P.-L., 1992, User's guide to viscosity solutions of second order partial differential equations. *Bulletin of the American Mathematical Society*, **27**, 1–67.
- DOYLE, J., GLOVER, K., KHARGONEKAR, P., and FRANCIS, B., 1988, State-space solutions to standard  $H_2$  and  $H_\infty$  control problems. *Proceedings of the American Control Conference*, Atlanta, GA.
- FLEMING, W. H., and RISHEL, R. W., 1975, *Deterministic and Stochastic Optimal Control* (Springer).
- FLEMING, W. H., and SONER, H. M., 1993, *Controlled Markov Processes and Viscosity Solutions* (Berlin: Springer).
- FLETCHER, C. A. J., 1984, *Computational Galerkin Methods*. Springer Series in Computational Physics (New York: Springer).
- FOSSEN, T. I., and SAGATUN, S. I., 1991, Adaptive control of nonlinear systems: A case study of underwater robotic systems. *Journal of Robotic Systems*, **8**, 393–412.
- FREEMAN, R. A., and KOKOTOVIC, P. V., 1995, Optimal nonlinear controllers for feedback linearizable systems. *Proceedings of the American Control Conference*, Seattle, Washington, pp. 2722–2726.
- GARRARD, W. L., and JORDAN, J. M., 1977, Design of nonlinear automatic flight control systems. *Automatica*, **13**, 497–505.
- GARRARD, W. L., ENNIS, D. F., and SNELL, S. A., 1992, Nonlinear feedback control of highly maneuverable aircraft. *International Journal of Control*, **56**, 799–812.
- GEORGES, D., 1995, Galerkin Methods for the Solution of Nonlinear Optimal Regulator and H-infinity Control Problems. *Proceedings of the European Control Conference*, Rome, Italy, pp. 2078–2083.
- GEORGES, D., 1996, Solutions of nonlinear optimal regulator and H-infinity control problems via Galerkin's method. *European Journal of Control*, **2**, 211–226.
- GLAD, S. T., 1984, On the gain margin of nonlinear and optimal regulators. *IEEE Transactions on Automatic Control*, **29**, 615–620.
- GLAD, S. T., 1987, Robustness of nonlinear state feedback—a survey. *Automatica*, **23**, 425–435.
- GLAD, T., 1985, Robust nonlinear regulators based on Hamilton–Jacobi theory and Lyapunov functions. *IEEE Control Conference*, Cambridge, pp. 276–280.
- GOH, C. J., 1993, On the nonlinear optimal regulator problem. *Automatica*, **29**, 751–756.
- GUENTHER, R. B., and LEE, J. W., 1988 *Partial Differential Equations of Mathematical Physics and Integral Equations* (Englewood Cliffs, NJ: Prentice-Hall).

- GUNTHER, J., BEARD, R., WILSON, J., OLIPHANT, T., and STIRLING, W., 1997, Fast nonlinear filtering via Galerkin's method. *Proceedings of the American Control Conference*, Albuquerque, NM.
- HUANG, J., and LIN, C.-F., 1995, Numerical approach to computing nonlinear  $h_\infty$  control laws. *AIAA Journal of Guidance, Control and Dynamics*, **18**, 989-994.
- ISIDORI, A., and ASTOLFI, A., 1992, Disturbance attenuation and  $H_\infty$ -control via measurement feedback in nonlinear systems. *IEEE Transactions on Automatic Control*, **37**, 1283-1293.
- KIRK, D. E., 1970, *Optimal Control Theory*. (Englewood Cliffs, NJ: Prentice-Hall).
- KUSHNER, H. J., 1990, Numerical methods for stochastic control problems in continuous time. *SIAM Journal on Control and Optimization*, **28**, 999-1048.
- LAWTON, J., and BEARD, R., 1998, Numerically efficient approximations to the Hamilton-Jacobi-Bellman equation. *American Control Conference*, Philadelphia, PA.
- LEAKE, R. J., and LIU, R.-W., 1967, Construction of suboptimal control sequences. *SIAM Journal on Control and Optimization*, **5**, 54-63.
- LEWIS, F. L., 1986, *Optimal Control* (New York: John Wiley & Sons).
- LU, P., 1993, A new nonlinear optimal feedback control law. *Control Theory and Advanced Technology*, **9**, 947-954.
- McLAIN, R. W., and BEARD, R. W., 1997, Nonlinear optimal control of a hydraulically actuated positioning system. *Proceedings of the ASME International Mechanical Engineering Congress and Exposition, Fluid Power Systems and Technology Division*, Dallas, TX.
- McLAIN, R. W., and BEARD, R. W., 1998b, Nonlinear optimal control of an underwater robotic vehicle. *International Conference on Robotics and Automation*, Leuven, Belgium.
- McLAIN, T. W., and BEARD, R. W., 1998a, Nonlinear optimal control design of a missile autopilot. *AIAA Guidance, Navigation and Control Conference*, Boston, MA.
- MIL'SHTEIN, G. N., 1964, Successive approximations for solution of one optimal problem. *Automation and Remote Control*, **25**, 298-306.
- REICHERT, R. T., 1990, Robust autopilot design using  $\mu$ -synthesis. *Proceedings of the American Control Conference*, pp. 2368-2373.
- REKASIUS, Z. V., 1964, Suboptimal design of intentionally nonlinear controllers. *IEEE Transactions on Automatic Control*, **9**, 380-386.
- ROBINETT, R. D., PARKER, G. G., SCHAUB, H., and JUNKINS, J. L., 1997, Lyapunov optimal saturated control for nonlinear systems. *Journal of Guidance, Control and Navigation*, **20**, 1083-1088.
- RYAN, E. P., 1984, Optimal feedback control of bilinear systems, *Journal of Optimization Theory and Applications*, **44**, 333-362.
- SAGE, A. P., and WHITE III, C. C., 1977, *Optimum Systems Control*, 2nd edn (Englewood Cliffs, NJ: Prentice-Hall).
- SARIDIS, G. N., and BALARAM, J., 1986, Suboptimal control for nonlinear systems. *Control Theory and Advanced Technology*, **2**, 547-562.
- SARIDIS, G. N., and LEE, C.-S. G., 1979, An approximation theory of optimal control for trainable manipulators. *IEEE Transactions on Systems, Man, and Cybernetics*, **9**, 152-159.
- SARIDIS, G. N., and WANG, F., 1994, Suboptimal control of nonlinear stochastic systems. *Control Theory and Advanced Technology*, **10**, 847-871.
- TSITSIKLIS, J. N., and ATHANS, M., 1984, Guaranteed robustness properties of multivariable nonlinear stochastic optimal regulators. *IEEE Transactions on Automatic Control*, **29**, 690-696.
- TZAFESTAS, S. G., ANAGNOSTOU, K. E., and PIMENIDES, T. G., 1984, Stabilizing optimal control of bilinear systems with a generalized cost. *Optimal Control Applications and Methods*, **5**, 111-117.
- VAISBORD, E. M., 1963, An approximate method for the synthesis of optimal control. *Automation and Remote Control*, **24**, 1626-1632.
- VAN DER SCHAFT, A. J., 1992,  $L_2$ -gain analysis of nonlinear systems and nonlinear systems and nonlinear state feedback  $H_\infty$  control. *IEEE Transactions on Automatic Control*, **37**, 770-784.
- WISE, K. A., and SEDWICK, J. L., 1994, Successive approximation solution of the HJI equation. *Proceedings of the 33rd Conference on Decision and Control*, Lake Buena Vista, FL, pp. 1387-1391.