

Successive Galerkin Approximation Algorithms for Nonlinear Optimal and Robust Control.

Randal W. Beard
Electrical Engineering Dept.
Brigham Young University
Provo, UT 84602
beard@ee.byu.edu

Timothy W. McLain
Mechanical Engineering Dept.
Brigham Young University
Provo, UT 84602
tmclain@et.byu.edu

Abstract

Nonlinear optimal control and nonlinear \mathcal{H}_∞ 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 worst, 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 problems have remained illusive. 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 computations 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}_∞ robust control [1]. Both of these paradigms extend naturally to nonlinear systems. Although the spaces \mathcal{H}_2 and \mathcal{H}_∞ are not used in the nonlinear setting, we will following the current nomenclature in the literature and refer to the corresponding nonlinear problems as nonlinear \mathcal{H}_2 control and nonlinear \mathcal{H}_∞ control. Theoretical results in nonlinear \mathcal{H}_2 control were developed in the late 1950's and 1960's and are contained in numerous textbooks (cf. [2, 3, 4]). The development of nonlinear \mathcal{H}_∞ control is more recent, with some of the seminal papers being [5, 6, 7].

For linear, state-feedback systems, both the \mathcal{H}_2 and \mathcal{H}_∞ control problems are solved by the solution to a matrix Riccati equation [8]. Since efficient numerical techniques exist for solving Riccati equations, the \mathcal{H}_2 and \mathcal{H}_∞ paradigms have been applied to a wide variety of problems, including robotics, flight control, satellite attitude control, and missile control, to name a few.

Although the theory for nonlinear \mathcal{H}_2 and \mathcal{H}_∞ is well developed, there have been very few applications, due to the lack of efficient numerical methods for solving these problems. To solve the nonlinear, state-feedback \mathcal{H}_2 optimal control problem requires the solution of the Hamilton-Jacobi-Bellman (HJB) equation [4]. To solve the nonlinear, state-feedback \mathcal{H}_∞ control problem requires the solution of the Hamilton-Jacobi-Isaacs (HJI) equation [5].

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 have been difficult to solve analytically, there has been a great deal of research directed toward approximating their solutions. This research can be divided into roughly four different categories, namely

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

The most straight forward 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 [9, 10]. Examples for the HJI equation are [11]. This method is often extended by using many open loop trajectories to train a neural networks or some other scheme that interpolates between open-loop control trajectories, cf. [12].

Series approximations of the HJB and HJI equation 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 [13, 14]. For the HJI equation see [15].

The basic idea behind “regularization” is to add terms in the cost function of nonlinear \mathcal{H}_2 control, that cancel out the nonlinearities in the systems. Examples include [16, 17, 18, 19].

The classical methods of finite difference and finite element solutions have also been used to approximate Hamilton-Jacobi equation. To make the method rigorous, the viscosity solution of HJB and HJI equations are approximated. See [20] for an introduction to viscosity solutions and [21, 22] for applications to control.

There are problems with each of the methods listed above. The method of characteristics results in open-loop control, and require a lot of 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. In addition, these controls 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, grows exponentially in 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 disad-

vantages 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 approximation,
- guarantee that the approximation error goes to zero as the order of 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 deals with the curse of dimensionality,
- be easy to tune,
- 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 PDE's. 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, and 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 importantly, the curse of dimensionality is still an issue. As will be explained in Section 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 in the size of the state, if a standard complete basis (like polynomial or sinusoid) is blindly used in the algorithm, i.e., no basis selection, the amount of memory required to compute and store the coefficients grows exponentially. To overcome this problem, particular basis functions must be chosen that 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.

The successive Galerkin approximation (SGA) algorithm for HJB equations first appeared in [23]. An overview of the algorithm and some initial applications appear in [24]. The convergence of the Galerkin approximation of the Generalized Hamilton-Jacobi-Bellman equation is shown in [25], and the convergence of the SGA algorithm for the HJB equation is shown in [26]. The algorithm has also been applied to several problems. In [27] the SGA algorithm was used to synthesize nonlinear optimal controls for a hydraulically actuated positioning systems. In [28] a nonlinear optimal control is

designed for a missile autopilot. Finally, in [29] 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 [30, 31].

This paper extends our previous work by developing a successive Galerkin approximation algorithm for the nonlinear \mathcal{H}_∞ 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}_∞ settings.

The paper is organized as follows. In Section 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 Section 4 we give four illustrative examples. The purpose of that section is not to analyze 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 Section 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 Section 2.3, Galerkin's method is used to approximate the Generalized Hamilton-Jacob equations. Combin-

ing the two methods results in systematic approximation algorithms for the HJB and HJI equations that are easily implemented on a computer and that 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 will 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 Hamilton-Jacobi-Bellman (HJB) equation [4]

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

with 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}_∞ Control

The nonlinear \mathcal{H}_∞ control problem is stated in terms of the \mathcal{L}_2 gain of a system. Consider the system

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

where x and u have the same dimension as above and $w \in \mathbb{R}^q$ 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 γ 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 γ if there exists some $0 \leq \tilde{\gamma} \leq \gamma$ such that equation (5) holds for $\tilde{\gamma}$.

The nonlinear \mathcal{H}_∞ (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_γ^* , has \mathcal{L}_2 -gain less than or equal to γ . As shown by van der Schaft [5], 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 Hamilton-Jacobi-Isaacs (HJI) equation

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

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

The main difference between the HJB and HJI equations is that the negative definite term $-gR^{-1}g^T$ in the HJB equation is replaced by the (in general) indefinite term $\left(\frac{1}{2\gamma^2}kP^{-1}k^T - gR^{-1}g^T\right)$. Another difference between the \mathcal{H}_2 and \mathcal{H}_∞ control problems is the gain parameter γ . For γ greater than some lower bound γ_{LB} the HJI equation admits a continuously differentiable solution V_γ [32], however for $\gamma < \gamma_{LB}$ there is no solution. We will exploit this fact at the end of Section 2.3 where we augment the SGA algorithm to find an approximation to the optimal nonlinear \mathcal{H}_∞ 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 [33] 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 [34] and has also been studied in [35, 36, 37, 38, 39, 40]. In the context of the HJI equation, van der Schaft [5] 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

$$\begin{aligned} \frac{\partial V^{*T}}{\partial x}(f + gu^*) + l + \|u^*\|_R^2 &= 0 \\ u^* &= -\frac{1}{2}R^{-1}g^T \frac{\partial V^*}{\partial x}. \end{aligned}$$

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 follows.

Algorithm 2.1 (HJB: Successive Approximation)

Let $u^{(0)}$ be an initial stabilizing control law for the system (1) with stability region Ω .

For $i = 0$ **to** ∞

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 $\{u^{(i)}\}_{i=0}^{\infty}$ [34, 39].

- 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 Ω .
- If $u^{(0)}$ stabilizes system (1) on Ω 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 [41, 42, 43, 44].
- The control $u^{(i+1)}$ has better performance than the control $u^{(i)}$.

The essence of Algorithm 2.1 is to reduce the HJB equation to an infinite sequence of linear partial differential equations of the form

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

where u is a known function of x . This equation is termed the Generalized Hamilton-Jacobi-Bellman (GHJB) equation, and 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 [45]. 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 Section 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}_∞ Control

The HJI equation (6) can be rewritten as

$$\begin{aligned} \frac{\partial V_\gamma^T}{\partial x}(f + gu_\gamma + kw_\gamma) + h^T h + \|u_\gamma\|_R^2 - \gamma^2 \|w_\gamma\|_P^2 &= 0, \\ u_\gamma &= -\frac{1}{2}R^{-1}g^T \frac{\partial V_\gamma}{\partial x}, \\ w_\gamma &= \frac{1}{2}P^{-1}k^T \frac{\partial V_\gamma}{\partial x}. \end{aligned}$$

Writing the HJI equation in this form, shows that the source of the non-linearity in the HJI equation is the dependence of u_γ and w_γ on the value function V_γ . 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 2.2 (HJI: Successive Approximation)

Let $u^{(0)}$ be an initial stabilizing control law for the system (4) ($w \equiv 0$) with stability region Ω .

For $i = 0$ **to** ∞

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

For $j = 0$ **to** ∞

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

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

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 2.3 *If for each (i, j) , the system*

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

is asymptotically stable on Ω , 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 Ω .

Proof: Let x_0 be any point in Ω . Since system (12) is asymptotically stable for the pair $(i, j+1)$ we have that

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

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 (9) and simplifying, and the third equality is obtained by substituting equation (10) 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). Q.E.D.

Lemma 2.4 *If for each i , the system*

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

is asymptotically stable on Ω , then

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

for each $x \in \Omega$, where V_γ 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 Ω .

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 [5]. Q.E.D.

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

Lemma 2.5 *If $u_\gamma^{(i)}(x)$ is asymptotically stabilizing on Ω , 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) we get that

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

which proves asymptotic stability since the system dynamics are observable through h . Q.E.D.

As with Algorithm 2.1, the essence of Algorithm 2.2 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 convenience of the reader, we give a brief summary of Galerkin's method in the specific context of this paper. For a more rigorous and complete treatment see [46, 47, 48]. 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 Ω . 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

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

In Section 2.3.1 we use Galerkin's method to approximate the GHJB equation and show that when this approximation is combined with Algorithm 2.1, the result is a practical algorithm for approximating nonlinear optimal control laws. Similarly, a practical algorithm for approximating nonlinear \mathcal{H}_∞ control laws is derived in Section 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 Ω . 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

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

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

$$\begin{aligned}
& \int_{\Omega} [\mathbf{c}_N^T \nabla \Phi_N (f + gu) + l + \|u\|_R^2] \Phi_N dx = 0 \\
\implies & \mathbf{c}_N^T \left[\int_{\Omega} \nabla \Phi_N (f + gu) \Phi_N dx \right] = - \int_{\Omega} (l + \|u\|_R^2) \Phi_N dx \\
\implies & \left[\int_{\Omega} \Phi_N (f + gu)^T \nabla \Phi_N^T dx \right] \mathbf{c}_N = - \int_{\Omega} (l + \|u\|_R^2) \Phi_N dx \\
\implies & \left[\left(\int_{\Omega} \Phi_N f^T \nabla \Phi_N^T dx \right) + \left(\int_{\Omega} \Phi_N u^T g^T \nabla \Phi_N^T dx \right) \right] \mathbf{c}_N = - \int_{\Omega} l \Phi_N dx - \int_{\Omega} \|u\|_R^2 \Phi_N dx \\
\implies & (\mathbf{A}_1^B + \mathbf{A}_2^B(u)) \mathbf{c}_N = \mathbf{b}_1^B + \mathbf{b}_2^B(u),
\end{aligned}$$

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 multiply.

At this point, it appears that to use this approximation in Algorithm 2.1 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):

$$\begin{aligned}
\hat{u}_N &= -\frac{1}{2} R^{-1} g^T \frac{\partial V_N}{\partial x} \\
&= -\frac{1}{2} R^{-1} g^T \sum_{j=1}^N c_j \frac{\partial \phi_j}{\partial x} \\
&= -\frac{1}{2} R^{-1} g^T \nabla \Phi_N^T \mathbf{c}_N.
\end{aligned}$$

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

$$\begin{aligned}
\mathbf{A}_2^B(\hat{u}_N) &= \int_{\Omega} \Phi_N \hat{u}_N^T g^T \nabla \Phi_N^T dx \\
&= -\frac{1}{2} \int_{\Omega} \Phi_N \mathbf{c}_N^T \nabla \Phi_N g R^{-1} g^T \nabla \Phi_N^T dx \\
&= -\frac{1}{2} \sum_{j=1}^N c_j \int_{\Omega} \Phi_N \frac{\partial \phi_j^T}{\partial x} g R^{-1} g^T \nabla \Phi_N^T dx \\
&= -\frac{1}{2} \sum_{j=1}^N c_j \mathbf{G}_j,
\end{aligned}$$

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 of 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 2.1 to produce the following algorithm for computing an approximation to the HJI equation.

Algorithm 2.6 (HJB: Successive Galerkin Approximation)

Let $u^{(0)}$ be an initial stabilizing control law for the system (1) with stability region Ω .

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** ∞

Let

$$\begin{aligned}\mathbf{A} &= \begin{cases} \mathbf{A}_1^B + \mathbf{A}_2^B(u^{(0)}) & i = 0 \\ \mathbf{A}_1^B - \frac{1}{2} \sum_{j=1}^{\infty} c_j^{(i-1)} \mathbf{G}_j & i > 0 \end{cases} \\ \mathbf{b} &= \begin{cases} \mathbf{b}_1^B + \mathbf{b}_2^B(u^{(0)}) & i = 0 \\ \mathbf{b}_1^B - \frac{1}{4} \sum_{j=1}^{\infty} c_j^{(i-1)} \mathbf{G}_j \mathbf{c}_{\gamma,N}^{(i-1)} & i > 0 \end{cases}\end{aligned}$$

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 2.6.

- The control $u_N^{(i)}$ is a feedback control law: Algorithm 2.6 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: digitally, analog hardware, etc..
- It has been shown in [26] that for N sufficiently large, the control $u_N^{(i)}$ is stabilizing on Ω (for all i) and is robust in the same sense as the optimal control u^* .
- It was also shown in [26] that as $N \rightarrow \infty$, $V_N^{(i)} \rightarrow V^*$ and $u_N^{(i)} \rightarrow u^*$.

2.3.2 Nonlinear \mathcal{H}_∞ 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 Ω , and that the set $\{\phi_j\}_{j=1}^\infty$ is a complete basis for the domain of the GHJI equation (14). Arguments identical to 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

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

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 2.2 to obtain the following approximation algorithm for the HJI equation.

Algorithm 2.7 (HJI: Successive Galerkin Approximation)

Let $u^{(0)}$ be an initial stabilizing control law for the system (4) ($w \equiv 0$) with stability region Ω .

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

For $i = 0$ **to** ∞

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** ∞

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

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

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

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

Update the Disturbance:

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

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 convergence of the Galerkin approximation to Generalized Hamilton-Jacobi equations has been shown in [25]. The proof of the overall convergence of Algorithm 2.7 is beyond the scope of the present paper and will appear in a subsequent work.

To solve the nonlinear \mathcal{H}_∞ optimal control problem, we need to find the smallest γ for which the HJI equation has a solution. An approximation to this problem can be found by embedding Algorithm 2.7 in a bisection search algorithm for the optimal γ , where the test on whether γ is valid or not is whether Algorithm 2.7 converges or not.

3 Mitigating the Curse of Dimensionality

The curse of dimensionality arises in two ways in Algorithms 2.6 and 2.7. 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 grow, 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

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

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

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

Now 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:

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

Note that with this partition, we must compute a total of $M^n = 3^2$ n -dimensional quadratures or, if l is separable and Ω 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 and are developed explicitly in [49].

The approach solves the problem of exponential growth in the number of computations required to implement the algorithm, at the expense of requiring that all system equations are sums of separable functions and that

Ω is rectangular. It does not, however, reduce the number of basis functions required for a complete basis. This problem is fundamental to approximation theory and is typically resolved through a careful, albeit heuristic, selection of approximating functions. We adopt this approach in Section 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}_∞ 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 [27]. 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, Ω , was determined from the physical constraints on the system

$$0 \leq P_L \leq P_s$$

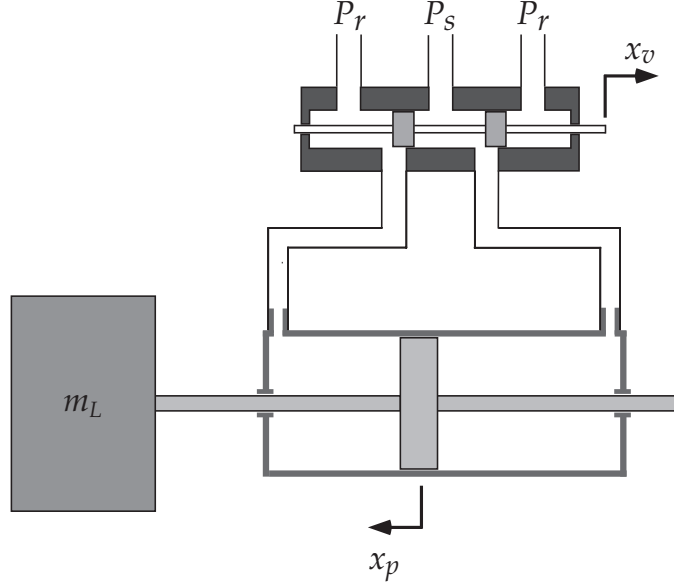


Figure 1: Hydraulic System Schematic

$$-Q_{max}/A \leq v_p \leq Q_{max}/A$$

$$x_{pmin} \leq x_p \leq x_{pmax}$$

$$x_{vmin} \leq x_v \leq x_{vmax}$$

where Q_{max} is the maximum rated no-load flow through the valve, x_{pmin} and x_{pmax} are bounds on the displacement of the piston, and x_{vmin} and x_{vmax} are bounds on the displacement of the valve spool.

For this problem, the basis functions were chosen to be

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

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. 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 $l_1 = P_L^2 + v_p^2 + e^2$ and the cost on the control set to $R = 1$. With these costs, the response of the system is slower than desired.

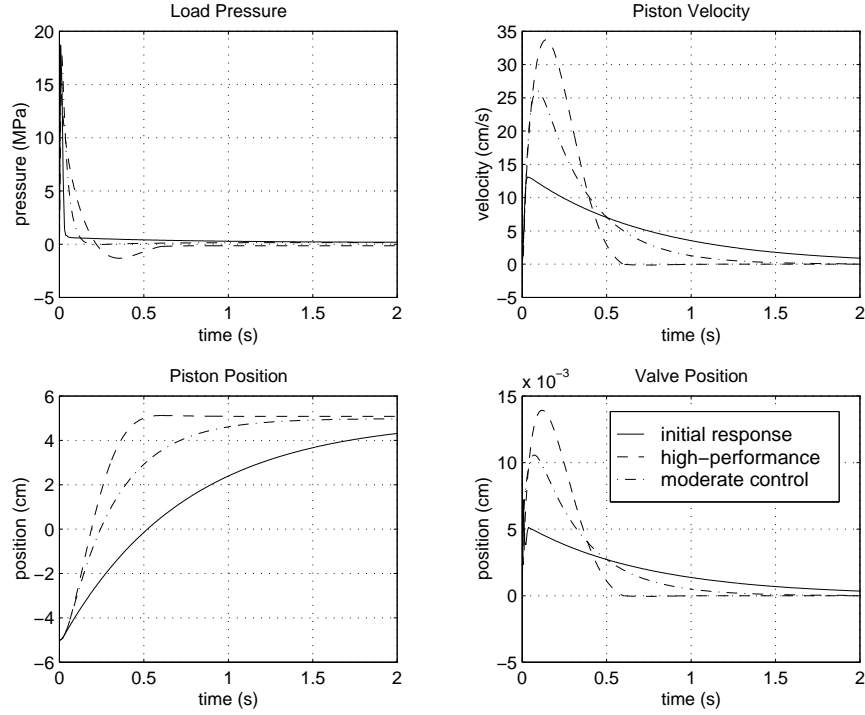


Figure 2: Optimal Control Responses — Hydraulic System

To improve the speed of response, the cost on the state errors was changed to $l_2 = 20P_L^2 + 0.2v_p^2 + 10e^2$. 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_2 was coupled with $R = 200$. It can be seen that less control is used while the response is slightly slower.

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 [28].

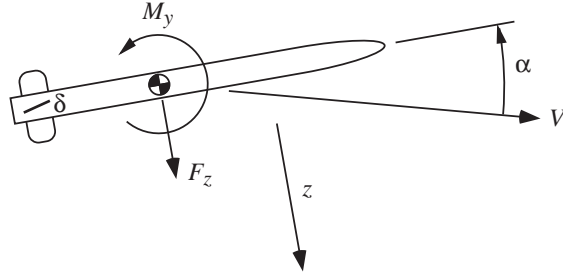


Figure 3: Missile System — Pitch Axis

The dynamic model for this system was taken from [50] and is represented by the following equations of motion

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

where q is the pitch rate, α is the angle of attack, δ is the tail fin deflection, and δ_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 η_z to a non-zero steady-state value. To do so, the states q , α , and δ , and the control δ_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 2.6, a change of variables must be performed to satisfy this requirement.

For a given desired steady-state acceleration η_{zd} , 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 $\eta_z \rightarrow \eta_{zd}$, $x \rightarrow 0$, and $u \rightarrow 0$ as $t \rightarrow \infty$, thereby enabling the application of Algorithm 2.6.

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

$$-200 \text{ deg/s} \leq x_1 \leq 200 \text{ deg/s}$$

$$-25 \text{ deg} \leq x_2 \leq 25 \text{ deg}$$

$$-800 \text{ deg/s} \leq x_3 \leq 800 \text{ deg/s}$$

$$-25 \text{ deg} \leq x_4 \leq 25 \text{ deg}.$$

A standard second-order set of basis functions (ϕ_1 through ϕ_{10}) was augmented with several other basis functions:

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

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. Figure 4 shows the responses obtained for each of the two controllers.

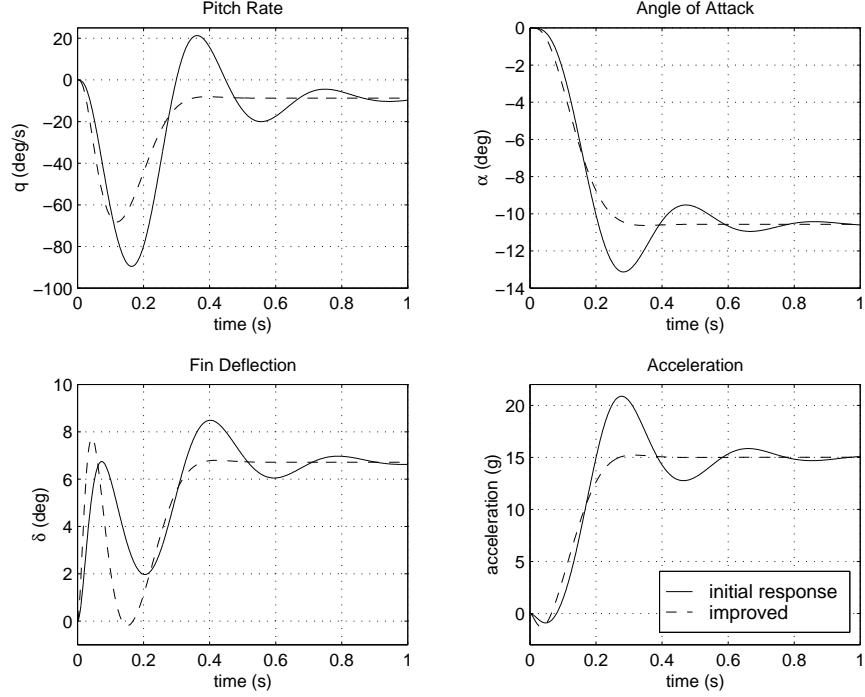


Figure 4: Missile System Response to $15g$ Acceleration Command

The first cost function was chosen to be the square of the difference between the desired and actual accelerations: $l = (\eta_{z_d} - \eta_z)^2$. 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, \hat{l} , 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: $\hat{l} = (\eta_{z_d} - \eta_z)^2 + 0.5x_1^2 + 25x_2^2$. As can be seen in Figure 4, the performance of missile is significantly improved with the new weighting function.

4.2 Nonlinear \mathcal{H}_∞ Control

4.2.1 Underwater Vehicle System

In this example, Algorithm 2.7 is used to synthesize a nonlinear \mathcal{H}_∞ controller to control the forward motion of a 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 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 due to currents or forces from an attached tether, τ 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 due to the difficulty in measuring them and due to the fact that 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 [51].

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

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

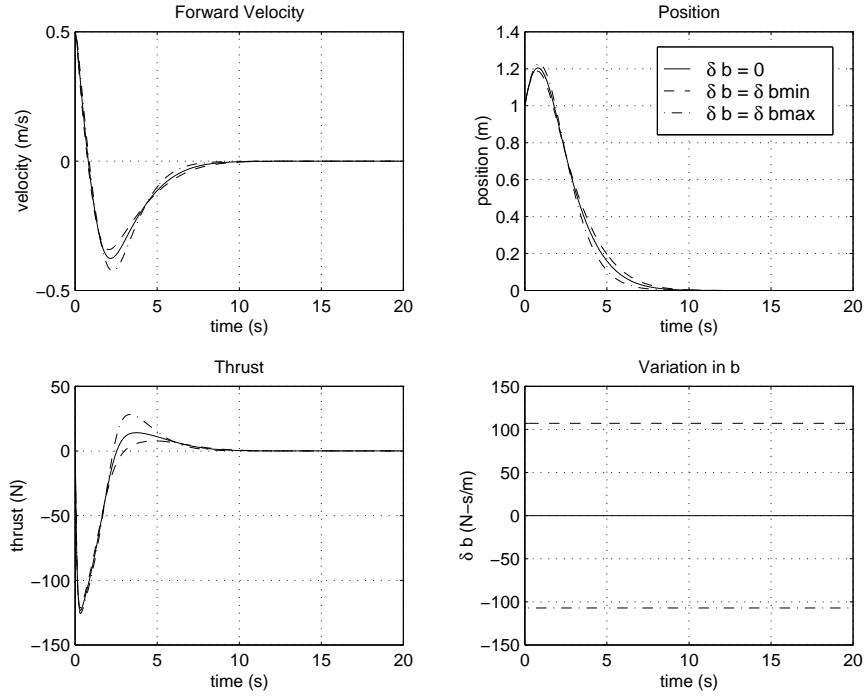


Figure 5: State Trajectories with Maximum Parametric Uncertainty

For the simulation results presented, station-keeping controllers were designed to regulate the velocity and position of the vehicle to zero. Figure 5 shows the response of the underwater vehicle system to initial conditions (0.5 m/sec 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, δb_{max} and δb_{min} . Though b is varied significantly, the similarity of results demonstrate 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

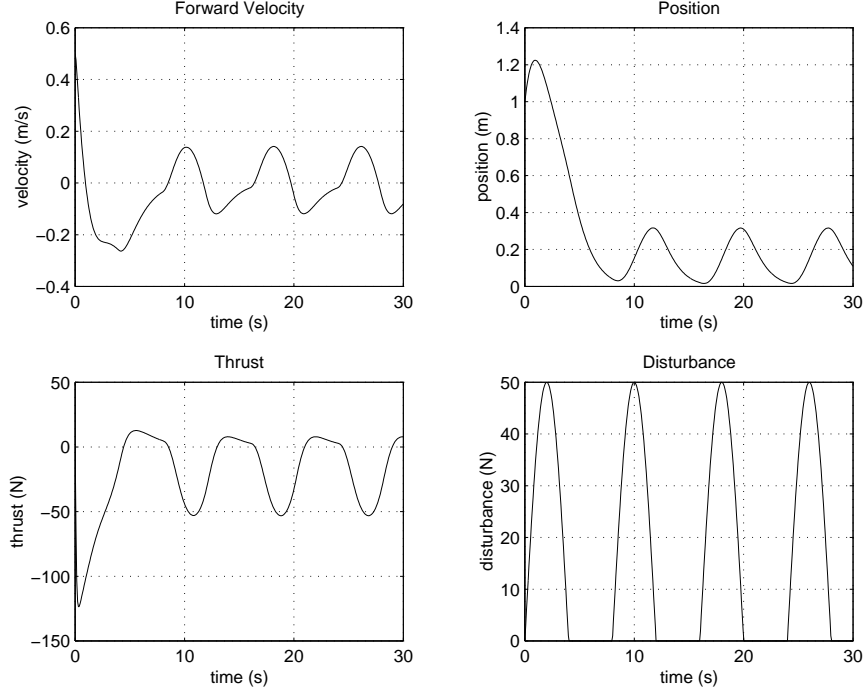


Figure 6: Tether Force Disturbance Rejection

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/sec and 1.0 m. These results demonstrate the ability of the controller to reject significant disturbances.

4.2.2 Hydraulic Actuation System

As a second example of nonlinear \mathcal{H}_∞ controller design using the Algorithm 2.7, 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 ρ . In practice, these parameters are often uncertain due to their sensitivity to changes in fluid temperature and due to 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

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

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

$$\begin{aligned} \frac{B}{\sqrt{\rho}} &= (1 + \mu) \frac{\hat{B}}{\sqrt{\hat{\rho}}} \\ B &= (1 + \beta) \hat{B}. \end{aligned}$$

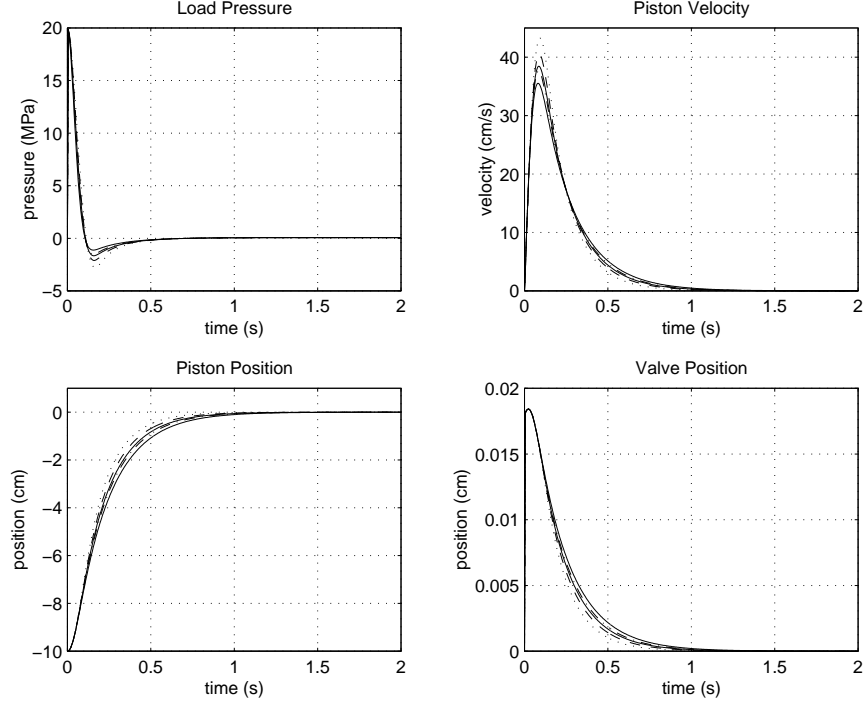


Figure 7: Hydraulic System Response with Maximum Parametric Uncertainty

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

$$y = P_L + v_p + x_p.$$

Figure 7 shows the results obtained using the nonlinear \mathcal{H}_∞ control law. The bulk modulus was allowed to vary $\pm 70\%$ from its nominal value, while the fluid density was allowed to vary $\pm 40\%$ from its nominal value. Such variations are extreme, but serve to demonstrate the robustness of the control achieved. 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}_∞ 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 are performed off-line and produce explicit expressions for feedback control laws, that 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 algorithm allow efficient approximations to the nonlinear (state-feedback) \mathcal{H}_2 optimal control and nonlinear (state-feedback) \mathcal{H}_∞ control problems, we believe that this paper represents a significant breakthrough in the control of nonlinear systems.

References

- [1] J. Doyle, K. Glover, P. Khargonekar, and B. Francis, "State-space solutions to standard H_2 and H_∞ control problems," in *Proceedings of the American Control Conference*, (Atlanta, GA), June 1988.
- [2] M. Athans and P. Falb, *Optimal Control*. New York, New York: McGraw-Hill Book Company, 1966.

- [3] W. H. Fleming and R. W. Rishel, *Deterministic and Stochastic Optimal Control*. Springer Verlag, 1975.
- [4] F. L. Lewis, *Optimal Control*. New York, New York: John Wiley and Sons, 1986.
- [5] A. J. van der Schaft, " L_2 -gain analysis of nonlinear systems and nonlinear systems and nonlinear state feedback H_∞ control," *IEEE Transactions on Automatic Control*, vol. 37, pp. 770–784, June 1992.
- [6] J. A. Ball, J. W. Helton, and M. L. Walker, " H_∞ control for nonlinear systems with output feedback," *IEEE Transactions on Automatic Control*, vol. 38, pp. 548–559, April 1993.
- [7] A. Isidori and A. Astolfi, "Disturbance attenuation and H_∞ -control via measurement feedback in nonlinear systems," *IEEE Transactions on Automatic Control*, vol. 37, pp. 1283–1293, September 1992.
- [8] S. Bittanti, A. Laub, and J. C. Willems, *The Riccati Equation*. New York, New York: Springer Verlag, 1991.
- [9] D. E. Kirk, *Optimal Control Theory*. Englewood Cliffs, New Jersey: Prentice-Hall, 1970.
- [10] A. P. Sage and C. C. White III, *Optimum Systems Control*. Englewood Cliffs, New Jersey: Prentice-Hall, 2nd ed., 1977.
- [11] K. A. Wise and J. L. Sedwick, "Successive approximation solution of the hji equation," in *Proceedings of the 33rd Conference on Decision and Control*, (Lake Buena Vista, FL), pp. 1387–1391, December 1994.
- [12] C. J. Goh, "On the nonlinear optimal regulator problem," *Automatica*, vol. 29, pp. 751–756, May 1993.
- [13] W. L. Garrard and J. M. Jordan, "Design of nonlinear automatic flight control systems," *Automatica*, vol. 13, pp. 497–505, 1977.
- [14] W. L. Garrard, D. F. Ennis, and S. A. Snell, "Nonlinear feedback control of highly maneuverable aircraft," *International Journal of Control*, vol. 56, no. 4, pp. 799–812, 1992.
- [15] J. Huang and C.-F. Lin, "Numerical approach to computing nonlinear h_∞ control laws," *AIAA Journal of Guidance, Control and Dynamics*, vol. 18, pp. 989–994, September-October 1995.

- [16] E. P. Ryan, "Optimal feedback control of bilinear systems," *Journal of Optimization Theory and Applications*, vol. 44, pp. 333–362, October 1984.
- [17] S. G. Tzafestas, K. E. Anagnostou, and T. G. Pimenides, "Stabilizing optimal control of bilinear systems with a generalized cost," *Optimal Control Applications and Methods*, vol. 5, pp. 111–117, 1984.
- [18] P. Lu, "A new nonlinear optimal feedback control law," *Control Theory and Advanced Technology*, vol. 9, pp. 947–954, December 1993.
- [19] R. A. Freeman and P. V. Kokotovic, "Optimal nonlinear controllers for feedback linearizable systems," in *Proceedings of the American Control Conference*, (Seattle, Washington), pp. 2722–2726, June 1995.
- [20] M. G. Crandall, H. Ishii, and P.-L. Lions, "User's guide to viscosity solutions of second order partial differential equations," *Bulletin of the American Mathematical Society*, vol. 27, pp. 1–67, July 1992.
- [21] H. J. Kushner, "Numerical methods for stochastic control problems in continuous time," *SIAM Journal on Control and Optimization*, vol. 28, pp. 999–1048, September 1990.
- [22] W. H. Fleming and H. M. Soner, *Controlled Markov Processes and Viscosity Solutions*. Berlin, Germany: Springer Verlag, 1993.
- [23] R. Beard, *Improving the Closed-Loop Performance of Nonlinear Systems*. PhD thesis, Rensselaer Polytechnic Institute, Troy, New York, 1995.
- [24] R. Beard, G. Saridis, and J. Wen, "Improving the performance of stabilizing control for nonlinear systems," *Control Systems Magazine*, vol. 16, pp. 27–35, October 1996.
- [25] R. Beard, G. Saridis, and J. Wen, "Galerkin approximation of the generalized Hamilton-Jacobi-Bellman equation," *Automatica*, 1997.
- [26] R. Beard, G. Saridis, and J. Wen, "Approximate solutions to the time-invariant Hamilton-Jacobi-Bellman equation," *Journal of Optimization Theory and Applications*, vol. 96, March 1998.
- [27] T. W. McLain and R. W. Beard, "Nonlinear optimal control of a hydraulically actuated positioning system," in *Proceedings of the ASME International Mechanical Engineering Congress and Exposition, Fluid Power Systems and Technology Division*, (Dallas, TX), November 1997.

- [28] T. W. McLain and R. W. Beard, "Nonlinear optimal control design of a missile autopilot," in *AIAA Guidance, Navigation and Control Conference*, (Boston, MA), 1998. (Submitted).
- [29] T. W. McLain and R. W. Beard, "Nonlinear optimal control of an underwater robotic vehicle," in *International Conference on Robotics and Automation*, 1998. (Submitted).
- [30] J. Gunther, R. Beard, J. Wilson, T. Oliphant, and W. Stirling, "Fast nonlinear filtering via Galerkin's method," in *Proceedings of the American Control Conference*, (Albuquerque, NM), 1997.
- [31] R. W. Beard, J. Gunther, J. Lawton, and W. Stirling, "The nonlinear projection filter," *AIAA Journal of Guidance, Control and Dynamics*, 1998. (Submitted).
- [32] T. Başar and P. Bernhard, *H_∞ -Optimal Control and Related Minimax Design Problems*. Birkhäuser, 1995.
- [33] R. E. Bellman, *Dynamic Programming*. Princeton, New Jersey: Princeton University Press, 1957.
- [34] G. N. Saridis and C.-S. G. Lee, "An approximation theory of optimal control for trainable manipulators," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, pp. 152–159, March 1979.
- [35] E. M. Vaisbord, "An approximate method for the synthesis of optimal control," *Automation and Remote Control*, vol. 24, pp. 1626–1632, 1963.
- [36] Z. V. Rekasius, "Suboptimal design of intentionally nonlinear controllers," *IEEE Transactions on Automatic Control*, vol. 9, no. 4, pp. 380–386, 1964.
- [37] G. N. Mil'shtein, "Successive approximations for solution of one optimal problem," *Automation and Remote Control*, vol. 25, pp. 298–306, 1964.
- [38] R. J. Leake and R.-W. Liu, "Construction of suboptimal control sequences," *SIAM Journal on Control and Optimization*, vol. 5, no. 1, pp. 54–63, 1967.
- [39] G. N. Saridis and J. Balaram, "Suboptimal control for nonlinear systems," *Control Theory and Advanced Technology*, vol. 2, pp. 547–562, September 1986.
- [40] G. N. Saridis and F. Wang, "Suboptimal control of nonlinear stochastic systems," *Control Theory and Advanced Technology*, vol. 10, pp. 847–871, December 1994.

- [41] S. T. Glad, "On the gain margin of nonlinear and optimal regulators," *IEEE Transactions on Automatic Control*, vol. 29, pp. 615–620, July 1984.
- [42] J. N. Tsitsiklis and M. Athans, "Guaranteed robustness properties of multivariable nonlinear stochastic optimal regulators," *IEEE Transactions on Automatic Control*, vol. 29, pp. 690–696, August 1984.
- [43] T. Glad, "Robust nonlinear regulators based on Hamilton-Jacobi theory and Lyapunov functions," in *IEE Control Conference*, (Cambridge), pp. 276–280, 1985.
- [44] S. T. Glad, "Robustness of nonlinear state feedback—a survey," *Automatica*, vol. 23, no. 4, pp. 425–435, 1987.
- [45] R. D. Robinett, G. G. Parker, H. Schaub, and J. L. Junkins, "Lyapunov optimal saturated control for nonlinear systems," *Journal of Guidance, Control and Navigation*, vol. 20, pp. 1083–1088, November-December 1997.
- [46] R. B. Guenther and J. W. Lee, *Partial Differential Equations of Mathematical Physics and Integral Equations*. Prentice-Hall, 1988.
- [47] M. A. Celia and W. G. Gray, *Numerical Methods for Differential Equations*. Prentice Hall, 1992.
- [48] C. A. J. Fletcher, *Computational Galerkin Methods*. Springer Series in Computational Physics, New York, New York: Springer Verlag, 1984.
- [49] J. Lawton and R. Beard, "Numerically efficient approximations to the Hamilton-Jacobi-Bellman equation," in *American Control Conference*, 1998. Preprint available at <http://www.ee.byu.edu/~beard/publications.html>.
- [50] R. T. Reichert, "Robust autopilot design using μ -synthesis," in *Proceedings of the American Control Conference*, pp. 2368–2373, 1990.
- [51] T. I. Fossen and S. I. Sagatun, "Adaptive control of nonlinear systems: A case study of underwater robotic systems," *The Journal of Robotic Systems*, vol. 8, no. 3, pp. 393–412, 1991.