
ELSEVIER

Available online at www.sciencedirect.com
SCIENCE DIRECT®

Computers and Mathematics with Applications 49 (2005) 1113–1126

An International Journal
computers & mathematics with applications

www.elsevier.com/locate/camwa

POD-Based Feedback Control of the Burgers Equation by Solving the Evolutionary HJB Equation

K. KUNISCH AND L. XIE
Karl-Franzens-Universität Graz, Institut für Mathematik
Heinrichstrasse 36, A-8010 Graz, Austria

(Received and accepted July 2004)

Abstract— A new approach for solving finite-time horizon feedback control problems for distributed parameter systems is proposed. It is based on model reduction by proper orthogonal decomposition combined with efficient numerical methods for solving the resulting low-order evolutionary Hamilton-Jacobi-Bellman (HJB) equation. The feasibility of the proposed methodology is demonstrated by means of optimal feedback control for the Burgers equation. The method for solving the HJB equation is first tested on several 1-D problems and then successfully applied to the control of the reduced order Burgers equation. The effect of noise is investigated, and parallelism is used for computational speedup.

Keywords— Dynamic programming, Hamilton-Jacobi-Bellman equation, Closed loop control, Proper orthogonal decomposition, Burgers equation.

1. INTRODUCTION

Optimal feedback control for distributed parameter systems is a significant challenge due to the computational complexity of the discretized HJB equation which still resides in a high-dimensional finite-dimensional space, if discretization methods based on, e.g., finite-element or finite-difference methods are chosen. Here, in contrast, we take the suboptimal feedback approach, i.e., first utilize a proper model reduction technique before addressing numerically the resulting (relatively) low-dimensional HJB equation. For a survey of suboptimal feedback strategies, we refer to [1,2], for instance. Typical model reduction techniques in the context of optimal control include balanced truncation [3, Chapter 7], reduced basis methods [4] and proper orthogonal decomposition, see, e.g., [5–8]. While balanced truncation is geared towards linear systems, the latter two reduction methods can equally well be used for nonlinear systems. Since proper orthogonal decomposition contains an efficient inherent mechanism for choosing the basis elements we use this approach in the present paper for model reduction.

The numerical treatment of the evolutionary HJB equation was treated by several authors. In [9], a small artificial diffusion terms is added to the HJB equation to make it elliptic and

Supported by the Fonds zur Förderung der wissenschaftlichen Forschung under SFB 03 "Optimierung und Kontrolle".

0898-1221/05/$ - see front matter
doi:10.1016/j.camwa.2004.07.022

the modified method of characteristics is applied to solve the resulting equation. In [10], an upwind FDM (finite-difference method) is explored. In this approach, ghost grids are introduced to realize an artificial boundary condition. In the present paper, a local Lax-Friedrichs scheme is combined with a total variation diminishing (TVD) Runge-Kutta time stepping method to solve the HJB equation.

2. HJB EQUATION FOR THE FINITE-HORIZON PROBLEM

We consider the optimal control problem

$$\min \quad J(x,t,u) = \int_0^t e^{-\mu s} L(y(s),u(s))\,\mathrm{d}s, \tag{2.1}$$

$$\begin{aligned} \text{subject to} \qquad \frac{d}{ds}y(s) &= F(y(s),u(s)), \quad 0 \le s \le T, \\ y(0) &= x, \\ u &\in \mathcal{U}, \end{aligned} \tag{2.2}$$

where $y : [0,T] \to \mathbb{R}^l$, $x \in \mathbb{R}^l$, $u : [0,T] \to \mathbb{R}^k$, $F : \mathbb{R}^l \times \mathbb{R}^k \to \mathbb{R}^l$, $L : \mathbb{R}^l \times \mathbb{R}^k \to \mathbb{R}$, $\mu \ge 0$, $t \in (0,T]$ and

$$\mathcal{U} = \left\{u : [0,T] \to \mathbb{R}^k : a \le u \le b\right\},$$

with $a \in \mathbb{R}^k$ and $b \in \mathbb{R}^k$ fixed bounds for the controls. The associated minimal value functional is defined by

$$v(x,t) = \inf_{u \in \mathcal{U}} J(x,t,u). \tag{2.3}$$

Under well-known conditions [11] the minimal value functional satisfies the dynamic programming principle (DPP)

$$v(x,t) = \inf_{u \in \mathcal{U}} \left\{ \int_0^\tau L\left(y\left(s;x,u\right),u(s)\right) e^{-\mu s}\,\mathrm{d}s + v\left(y\left(\tau;x,u\right),t-\tau\right) e^{-\mu\tau} \right\}, \tag{DPP}$$

for every $\tau \in (0,t]$, where $y(\cdot;x,u)$ denotes the solution to (2.2) as a function of the initial solution $x \in \mathbb{R}^l$ and the control $u \in \mathcal{U}$. If v is C^1, the following HJB equation can be derived from (DPP):

$$\frac{\partial v(x,t)}{\partial t} + \mu v(x,t) + \sup_{u \in U} \left(- F^T(x,u)\nabla v(x,t) - L(x,u)\right) = 0, \tag{2.4a}$$

$$v(x,0) = 0, \tag{2.4b}$$

where $U = \{u \in \mathbb{R}^k : a \le u \le b\}$, $(x,t) \in \mathbb{R}^l \times \mathbb{R}^+$, and ∇ denotes the gradient with respect to x. If v is merely continuous then it can be proved [11, Chapter 3] that it is the unique viscosity solution to the HJB equation. We shall not enter into details of DPP and HJB, but rather refer to [11], for example.

Solving (2.4) is unfeasible for dynamical systems in higher-dimensional spaces as they typically arise from standard discretization of partial differential equations. We shall address this issue in the following sections. Here we address the sup-operation in (2.4a) which can be simplified if F and L are separable, more precisely if

$$F(x,u) = f(x) + Bu \qquad \text{and} \qquad L(x,u) = l(x) + \frac{\beta}{2}u^T u, \tag{2.5}$$

where $B \in \mathbb{R}^{l \times k}$ and $\beta > 0$. Then, given x, t, and v, the supremum is characterized by the argument $u \in U$ satisfying

$$\left(B^T\nabla v + \beta u, w - u\right)_{\mathbb{R}^k} \ge 0, \qquad \text{for all } w \in U.$$

This is equivalent to

$$u = P_U\left(-\frac{1}{\beta}B^T\nabla v\right), \tag{2.6}$$

where P_U denote the projection in $\mathbb{R}^k$ onto U. Substituting (2.6) into (2.4a) yields

$$\frac{\partial v}{\partial t} - \left(f + BP_U\left(-\frac{1}{\beta}B^T\nabla v\right)\right)^T \nabla v - l - \frac{\beta}{2}\left(P_U\left(-\frac{1}{\beta}B^T\nabla v\right)\right)^T\left(P_U\left(-\frac{1}{\beta}B^T\nabla v\right)\right) = 0,$$

where the dependence on (x,t) is suppressed. We next turn to the feedback synthesis, i.e., the construction of the optimal control $u^*(t)$ in terms of the current state $y^*(t)$. To heuristically derive the feedback representation, let (y^*,u^*) denote an optimal trajectory-optimal control pair on $[0,T]$. Then from (DPP) we obtain for $t \in (0,T]$

$$\begin{aligned}\frac{\partial}{\partial t}v\left(y^*(t),T-t\right) + \mu v\left(y^*(t),T-t\right)& \\ + \left(-F^T\left(y^*(t),u^*(t)\right)\nabla v\left(y^*(t),T-t\right) - L\left(y^*(t),u^*(t)\right)\right) = 0,& \\ v\left(y^*(T),0\right) = 0,&\end{aligned} \tag{2.7}$$

which implies the feedback systhesis

$$u^*(t) = \underset{u\in U}{\operatorname{argmax}}\left\{-F^T\left(y^*(t),u\right)\nabla v\left(y^*(t),T-t\right) - L\left(y^*(t),u\right)\right\},$$

and in the separable case (2.5)

$$u^*(t) = P_U\left(-\frac{1}{\beta}B^T\nabla v\left(y^*(t),T-t\right)\right). \tag{2.8}$$

3. LOCAL LAX-FRIEDRICHS SCHEME AND TIME STEPPING

In this section, we describe the procedure that we used to solve the time dependent HJB equation. For the spatial discretization we employed a third-order local Lax-Friedrichs scheme. For simplicity of notation we describe the scheme for one-dimensional problems. The extension to the higher dimensions is straightforward.

Let $v_k = v(x_k)$ denote a grid function defined on a uniform grid x_k with step length Δx. The first order forward and backward finite-difference approximations to the differentiation operator at x_k are denoted by

$$v^+_{x,k} = \frac{v_{k+1}-v_k}{\Delta x}, \qquad v^-_{x,k} = \frac{v_k - v_{k-1}}{\Delta x}; \tag{3.1}$$

as well as third order approximations,

$$\begin{aligned}\nabla^- v &= \frac{1}{3}v^+_{x,k} + \frac{5}{6}v^+_{x,k-1} - \frac{1}{6}v^+_{x,k-2},\\ \nabla^+ v &= \frac{1}{3}v^+_{x,k-1} + \frac{5}{6}v^+_{x,k} - \frac{1}{6}v^+_{x,k+1}.\end{aligned} \tag{3.2}$$

The semidiscretized form for the HJB equation $v_t + \mu v + H(\nabla v) = 0$ is constructed by the local Lax-Friedrichs (LLF) scheme [12],

$$v_t + \mu v = -H\left(\frac{1}{2}\left(\nabla^+ v + \nabla^- v\right)\right) + \mathcal{A}\left(\nabla^+ v, \nabla^- v\right)\cdot\frac{\left(\nabla^+ v - \nabla^- v\right)}{2}, \tag{3.3}$$

where

$$\mathcal{A}\left(\nabla^+ v, \nabla^- v\right) = \max\left|\hat{H}_{\nabla v}\right|,$$

with $\hat{H}_{\nabla v}$ denoting the derivative of H with respect to ∇v, and the maximum taken over $I(\nabla^- v, \nabla^+ v) = [\min(\nabla^- v, \nabla^+ v), \max(\nabla^- v, \nabla^+ v)]$. The mechanism of the LLF scheme determines the numerical dissipation based on local quantities, without introducing an excessive amount of artificial viscosity. Therefore, as compared to the common Lax-Friedrichs scheme, the LLF scheme is efficient in capturing discontinuities with higher resolution.

It is well-known that non-TVD time stepping schemes can generate nonphysical oscillations even if the spatial discretization is TVD. Therefore a TVD time stepping scheme is necessary for the temporal discretization. In this paper, we employ a three-stage third-order TVD Runge-Kutta scheme [13,14]. Expressing the evolution problem in the form $v_t = L(v)$, this results in:

$$\begin{aligned} v_{(1)} &= v_n + \Delta t L(v_n), \\ v_{(2)} &= \frac{3}{4} v_n + \frac{1}{4}(v_{(1)} + \Delta t L(v_{(1)})), \\ v_{n+1} &= \frac{1}{3} v_n + \frac{2}{3}(v_{(2)} + \Delta t L(v_{(2)})). \end{aligned}$$

The time step Δt should be chosen, such that

$$\frac{\Delta t}{\min\left(\Delta x / (|f| + |Bu|)\right)} < \mathrm{CFL}, \tag{3.4}$$

where CFL is the Courant-Friedrichs-Lewy (CFL) coefficient. This TVD Runge-Kutta scheme is an optimal three-stage three-order scheme [13] in the sense that the associated CFL is equal to 1, which indicates that a fully exploited time step can be taken in the time discretization.

We arrive at the following algorithm to solve the evolutionary HJB equation.

Step 1. Set $v_0 = 0$, $i = 0$.
Step 2. Calculate $\nabla v_i^{\pm}$, and $u_i = P_U(-(1/\beta) B^T \nabla v)$.
Step 3. Calculate the RHS of (3.3);
and obtain v_{i+1} via Runge-Kutta time stepping.
Step 4. If the final time reached, then stop;
if not set $i = i + 1$ and go back to Step 2.

4. POD-BASED REDUCED-ORDER METHOD

In this section, we briefly describe system reduction of infinite-dimensional control systems by proper orthogonal decomposition. For details we refer to [7,15], for example.

4.1. The Abstract Dynamical System

Let V and H be real Hilbert spaces with V densely embedded in H. By $\langle \cdot, \cdot \rangle_H$ we denote the inner product in H and analogously for V. The inner product in V is given by a symmetric bounded, coercive, bilinear form $a : V \times V \to \mathbb{R}$:

$$\langle \varphi, \psi \rangle_V = a(\varphi, \psi), \qquad \text{for all } \varphi, \psi \in V. \tag{4.1}$$

For $T \in (0, \infty)$ let $L^2(0,T;V)$ denote the usual Lebesgue space of square integrable V-valued functions, and define

$$W(0,T;V) = \left\{ \varphi \in L^2(0,T;V) : \varphi_t \in L^2\left(0,T;V^*\right) \right\},$$

where V^* denotes the dual space of V. Furthermore, let $N : V \to V^*$ be a nonlinear mapping, let $\hat{U}$ denote the Hilbert space of controls, and let $B \in \mathcal{L}(\hat{U}, V^*)$ be the control operator. Given $y_\circ \in H$ and $u \in L^2(0,T;\hat{U})$ the abstract control system is given by

$$\frac{\mathrm{d}}{\mathrm{d}t} \langle y(t), \varphi \rangle_H + a(y(t), \varphi) + \langle N(y(t)), \varphi \rangle_{V^*,V} = \langle B(u(t)), \varphi \rangle_{V^*,V}, \tag{4.2a}$$

for all $\varphi \in V$ and

$$y(0) = y_\circ. \tag{4.2b}$$

Conditions which guarantee the existence of a unique solution $y = y(\cdot; y_\circ, u)$ are well known from the literature, see e.g., [16,17].

4.2. The POD Reduction

Given $n \in \mathbb{N}$, let

$$0 = t_0 < t_2 < \cdots < t_n < \infty \tag{4.3}$$

denote a grid in the interval $[0, T]$, and suppose that the *snapshots* $y_j = y(t_j)$ of (4.2) are available. These can be computed by an independent finite-element scheme, for example. We set

$$\mathcal{V} = \text{span}\,\{y_0, \ldots, y_n\} \subset V.$$

Let $\{\psi_i\}_{i=1}^d$ denote an orthonormal basis of $\mathcal{V}$ with $d = \dim \mathcal{V}$. Then each element of the $\mathcal{V}$ can be expressed as

$$y_j = \sum_{i=1}^{d} \langle y_j, \psi_i \rangle_H \psi_i, \qquad \text{for } j = 0, \ldots, n. \tag{4.4}$$

The POD method consists in choosing an orthonormal basis such that for every $\ell \in \{1, \ldots, d\}$ the mean square error between the elements y_j, $0 \le j \le n$, and the corresponding ℓ^{th} partial sum of (4.4) is minimized on average:

$$\begin{aligned} \min \quad & \mathfrak{J}(\psi_1, \ldots, \psi_\ell) = \sum_{j=0}^{n} \left\| y_j - \sum_{i=1}^{\ell} \langle y_j, \psi_i \rangle_H \psi_i \right\|_H^2 \\ \text{subject to} \quad & \langle \psi_i, \psi_j \rangle_H = \delta_{ij}, \qquad \text{for } 1 \le i \le \ell, \quad 1 \le j \le i. \end{aligned} \tag{4.5}$$

A solution $\{\psi_i\}_{i=1}^\ell$ to (4.5) is called *POD basis of rank* ℓ. The subspace spanned by the first ℓ POD basis functions is denoted by V^ℓ, i.e.,

$$V^\ell = \text{span}\,\{\psi_1, \ldots, \psi_\ell\}. \tag{4.6}$$

The solution of (4.5) is characterized by the necessary optimality condition which can be written as an eigenvalue problem. Let $\mathcal{Y}_n : \mathbb{R}^{n+1} \to X$ defined by

$$\mathcal{Y}_n v = \sum_{j=0}^{n} v_j y_j, \qquad \text{for } v \in \mathbb{R}^{n+1}, \tag{4.7}$$

with the adjoint $\mathcal{Y}_n^* : X \to \mathbb{R}^{n+1}$ given by

$$\mathcal{Y}_n^* z = (\langle z, y_0 \rangle_H, \ldots, \langle z, y_n \rangle_H)^\top, \qquad \text{for } z \in H. \tag{4.8}$$

Using a Lagrangian framework the optimality condition for (4.5) is given by

$$\mathcal{R}\psi = \lambda\psi, \tag{4.9}$$

where $\mathcal{R}z = \mathcal{Y}_n \mathcal{Y}^* z = \sum_{j=0}^n \langle z, y_j \rangle_H y_j$, for $z \in H$. By Hilbert-Schmit theory there exists an orthonormal basis $\{\psi_i\}_{i=0}^\infty$ for H and a sequence $\{\lambda_i\}_{i=0}^\infty \in \mathbb{R}^+$, such that

$$\mathcal{R}_n \psi_i = \lambda_i \psi_i, \quad \lambda_1 \ge \cdots \ge \lambda_d > 0, \qquad \text{and} \qquad \lambda_i = 0, \quad \text{for } i > d. \tag{4.10}$$

Moreover, $\mathcal{V} = \text{span}\,\{\psi_i\}_{i=1}^d$. Note that $\{\lambda_i\}_{i=0}^\infty$ as well as $\{\psi_i\}_{i=0}^\infty$ depend on n. Contents permitting the notation of this dependence is dropped.

The sequence $\{\psi_i\}_{i=1}^\ell$ solves the optimization problem (4.5) and is a POD basis of rank $\ell \le d$. In addition, we have the error formula

$$\mathfrak{J}(\psi_1, \ldots, \psi_\ell) = \sum_{j=0}^{n} \left\| y_j - \sum_{i=1}^{\ell} \langle y_j, \psi_i \rangle_H \psi_i \right\|_H^2 = \sum_{i=\ell+1}^{d} \lambda_i. \tag{4.11}$$

To obtain the POD approximation to (4.2a) we make the finite-dimensional ansatz

$$y^\ell(t,x) = \sum_{i=1}^{\ell} \mathrm{y}_i(t)\psi_i(x). \tag{4.12}$$

Here we do not discretize the control variable. In our application to the Burgers equation we consider boundary control, and hence, $\hat{U}$ is finite dimensional.

We introduce the mass and stiffness matrices by

$$\begin{aligned} \mathrm{M} &= ((\mathrm{m}_{ij})) \in \mathbb{R}^{\ell\times\ell}, \qquad \text{with } \mathrm{m}_{ij} = \langle \psi_j, \psi_i\rangle_H, \\ \mathrm{S} &= ((\mathrm{s}_{ij})) \in \mathbb{R}^{\ell\times\ell}, \qquad \text{with } \mathrm{s}_{ij} = a(\psi_j, \psi_i), \end{aligned}$$

the discretization of the nonlinear mapping $\mathrm{N}: \mathbb{R}^\ell \to \mathbb{R}^\ell$

$$(\mathrm{y}_1, \ldots, \mathrm{y}_\ell) \mapsto \mathrm{N}(\mathrm{y}_1, \ldots, \mathrm{y}_\ell) = (\mathrm{n}_i) \in \mathbb{R}^\ell, \qquad \text{with } \mathrm{n}_i = \left\langle N\left(\sum_{j=1}^{\ell} \mathrm{y}_i\psi_j\right), \psi_i \right\rangle_{V^*,V},$$

as well as the control input $\mathrm{b}: U \in \mathbb{R}^\ell$ by

$$u \mapsto \mathrm{b}(u) = (\mathrm{b}(u)_i) \in \mathbb{R}^\ell, \qquad \text{with } b(\mathrm{u})_i = \langle Bu, \psi_i\rangle_H.$$

The modal coefficients of the initial condition $y_0 \in \mathbb{R}^\ell$ are given by $(\mathrm{y}_\circ)_i = \langle y_\circ, \psi_i\rangle_H$. The solution vector of the reduced dynamical system is denoted by $\mathrm{y}^\ell(t) \in \mathbb{R}^\ell$. It is a solution of the POD-reduced order model given by

$$\begin{aligned} \dot{\mathrm{y}}^\ell(t) &= F(\mathrm{y}^\ell(t), u(t)), \qquad \text{for } t > 0, \\ \mathrm{y}^\ell(0) &= \mathrm{y}_\circ, \end{aligned} \tag{4.13}$$

where

$$F\left(\mathrm{y}^\ell, u\right) = \mathrm{M}^{-1}\left(-\mathrm{S}\mathrm{y}^\ell - \mathrm{N}\left(\mathrm{y}^\ell\right) + \mathrm{b}(u)\right).$$

6. 1-D TEST PROBLEMS

To test the efficiency of the proposed method for solving the HJB equation, we first solved several 1-D examples. The first test problem is the 1-D Hamilton-Jacobi equation

$$\phi_t + \frac{1}{4}\left(\phi_x^2 - 1\right)\left(\phi_x^2 - 4\right) = 0, \qquad \phi(x,0) = -2|x|. \tag{5.1}$$

The numerical solutions in the interval $[-1,1]$ at $T = 1$ are presented in Figure 1 for a series of grid systems. The larger the number of the grid points, the better the numerical solution approaches the exact solution which achieves -1 at $x = 0$ [18]. The nonsmoothness in the initial condition at $x = 0$ is propagated to the solution at later time instances. As shown in the figure, the nonsmoothness is well resolved via the proposed numerical scheme.

Then, we will solve a 1-D optimal control problem [9]

$$\min \quad \frac{\alpha}{2}\int_0^1 u^2(t)\,\mathrm{d}t - x(1), \tag{5.2a}$$

$$\text{subject to} \quad \dot{x}(t) = x(t)u(t), \tag{5.2b}$$

$$x(0) = y,$$

$$u(t) \in U = [-1,1].$$

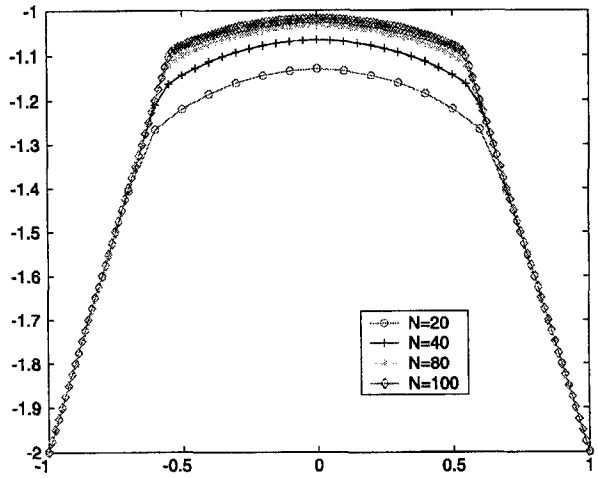

Figure 1. Solution to the 1-D HJ equation at $T = 1$.

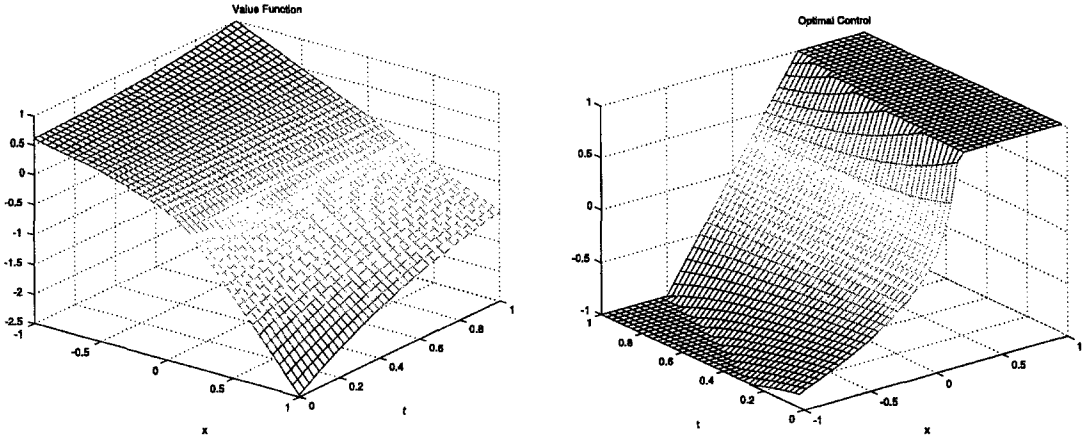

Figure 2. Value function (left) and the corresponding optimal control (right), $\alpha = 1/2$.

The corresponding HJB equation is

$$-v_t + \sup_{-1\le u\le 1}\left(-xuv_x - \frac{\alpha}{2}u^2\right) = 0,$$
$$v(T,x) = -x.$$

Here, we reversed the time direction in the HJB equation, as is frequently done for finite-horizon optimal control problems. The initial value problem is solved for $-1 \le x \le 1$ within the time horizon $(0,1]$. The computational results for $\alpha = 1/2$ are plotted in the Figure 2. The value function is smooth in the whole domain of interest, even though the optimal control is not.

The third test problem is the same as the second one, except that the admissible set for the control is $U = [0,1]$ and $\alpha = 0.001$. This case can be viewed as an asymptotic problem to the control problem

$$\begin{aligned} \min \quad & -x(1), \\ \text{s.t.} \quad & \dot{x}(t) = x(t)u(t), \\ & x(0) = y, \\ & u(t) \in U = [0,1], \end{aligned} \tag{5.3a}$$

which has the exact solution

$$v(t,x) = \begin{cases} -xe^{1-t}, & \text{if } x > 0, \\ -x, & \text{if } x \le 0. \end{cases} \tag{5.4}$$

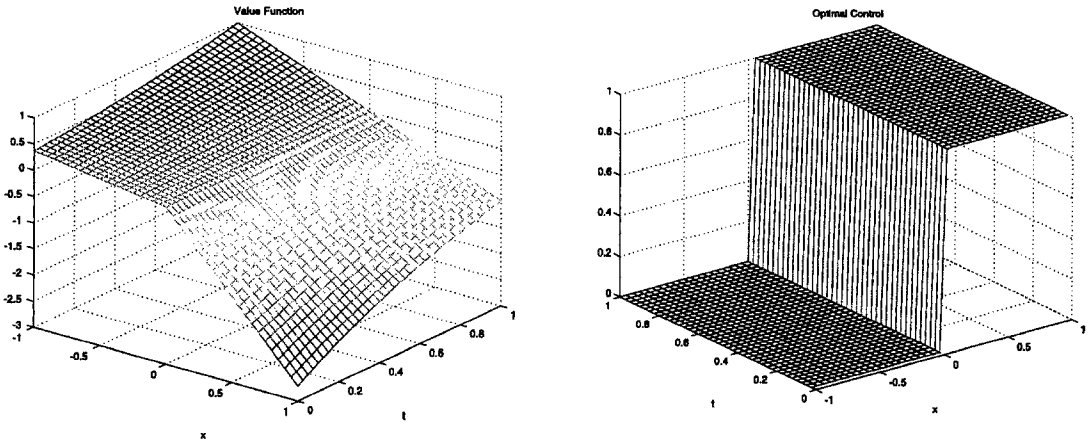

Figure 3. Value function (left) and the corresponding optimal control (right), $\alpha = 0.001$.

Figure 3 shows the value funtion and the corresponding optimal control. The value function is now nonsmooth at $x = 0$. At this location the optimal control jumps from one bound to the other.

In all the 1-D test cases we only employed the first-order backward and forward-difference approximations to the gradient of the value function. The computational results suggest that this is acceptable. Unfortunately, this is not the case for suboptimal feedback control for the Burgers equation. We will return to this issue later on.

6. APPLICATION TO SUBOPTIMAL CONTROL FOR THE BURGERS EQUATION

In this section, the proposed method will be employed to carry out the feedback design for the Burgers equation. We begin with the optimal control problem for the Burgers equation; then the POD technique is carried out to obtain a 4-D reduced order control problem, which is of the same form as described in Section 2 and can be solved by the proposed method. Finally the computational results are presented and discussed.

6.1. The Original Optimal Control Problem

Define the domains $\Omega = (0,1) \subset \mathbb{R}$, $Q = (0,T] \times \Omega$ and $\Sigma = (0,T] \times \partial\Omega$. In the context of Section 4.1, we set $H = L^2(\Omega)$, $V = H^1(\Omega)$, and define

$$a(\varphi,\phi) = \nu \int_\Omega \varphi'\phi' \, dx, \qquad \text{for } \varphi,\phi \in V,$$

with $\nu > 0$ and $B \in \mathcal{L}(\mathbb{R}, V^*)$ by

$$\langle Bu, \phi\rangle_{V^*,V} = u\,\phi(0).$$

For fixed $u_a \le u_b$, we set $U_{\text{ad}} = \{u \in \mathbb{R} : u_a \le u \le u_b\}$ and define the set of admissible controls

$$\mathcal{U}_{\text{ad}} = \{u \in L^2(0,T] : u(t) \in U_{\text{ad}}, \text{ for almost all } t \in (0,T]\}. \tag{6.1}$$

For a control $u \in \mathcal{U}_{\text{ad}}$ we consider the viscous Burgers equation

$$y_t - \nu y_{xx} + y y_x = 0, \qquad \text{in } Q, \tag{6.2a}$$

$$\nu y_x(\cdot,0) + \sigma_0 y(\cdot,0) = u, \qquad \text{in } (0,T], \tag{6.2b}$$

$$\nu y_x(\cdot,1) + \sigma_1 y(\cdot,1) = g, \qquad \text{in } (0,T], \tag{6.2c}$$

$$y(0,\cdot) = y_o, \qquad \text{in } \Omega, \tag{6.2d}$$

where $y_o \in L^2(\Omega)$ is a given initial condition, and σ_0, σ_1, g are real numbers. Henceforth, we consider weak solutions $y \in W(0,T;V)$ of (6.2) satisfying (6.2d) and

$$\begin{aligned} &\langle y_t(t), \varphi\rangle_{V^*,V} + \sigma_1 y(t,1)\varphi(1) - \sigma_0 y(t,0)\varphi(0) \\ &+a(y,\varphi) + \int_\Omega y(t)y'(t)\varphi \, dx = g\varphi(1) - u\varphi(0), \end{aligned} \tag{6.3}$$

for all $\varphi \in H^1(\Omega)$ and $t \in (0,T]$ a.e. For the functional analytic treatment of (6.2), we refer to [17,19], for example. We shall consider the cost functional

$$J(y,u) = \int_0^T \left(\frac{1}{2} \int_\Omega |y(t,x) - z(x)|^2 \, dx + \frac{\beta}{2} |u(t)|^2 \right) e^{-\mu t} \, dt,$$

where $z \in L^2(\Omega)$ is a given desired state, and $\mu \geq 0, \beta > 0$ are constants.

The optimal control problem is given by

$$\min J(y,u), \text{ such that } (y,u) \in W(0,T;V) \times \mathcal{U}_{ad} \text{ satisfies (6.2)} \tag{$\tilde{\mathrm{P}}$}$$

as a weak solution. It is straightforward to argue the existence of an optimal control for problem ($\tilde{\mathrm{P}}$), e.g., see [20].

6.2. POD-Based Reduced Order Model

Suppose that we have computed a POD basis utilizing, e.g., a finite-element code for the viscous Burgers equation and determined the basis functions as described in Section 4.2. To compute a POD solution of ($\tilde{\mathrm{P}}$) we make the ansatz for the state variable. In addition to the matrices and vectors defined in Section 4.2 we introduce the tensor

$$\mathrm{T} = (((\mathrm{b}_{ijk}))) \in \mathbb{R}^{\ell \times \ell \times \ell}, \qquad \text{with } \mathrm{b}_{ijk} = \int_\Omega \psi_j \psi_k' \psi_i dx,$$

and the vectors for the boundary conditions

$$\mathrm{d} = (\mathrm{d}_i) \in \mathbb{R}^\ell, \quad \text{with } \mathrm{d}_i = \psi_i(0), \qquad \mathrm{e} = (\mathrm{e}_i) \in \mathbb{R}^\ell, \quad \text{with } \mathrm{e}_i = \psi_i(1).$$

Then the nonlinear mapping $F : \mathbb{R}^\ell \times \mathbb{R} \to \mathbb{R}^\ell$ is given by

$$F\left(\mathrm{y}^\ell, u\right) = \mathrm{M}^{-1} \left(\left(-\mathrm{S} - \left(\mathrm{T} : \mathrm{y}^\ell\right)\right) \mathrm{y}^\ell + \mathrm{d} \left(\mathrm{d}^\mathsf{T} \sigma_0 \mathrm{y}^\ell - u \right) - \mathrm{e} \left(\mathrm{e}^\mathsf{T} \sigma_1 \mathrm{y}^\ell - g \right) \right).$$

The value function v, defined for any initial state $\mathrm{y}_o \in \mathbb{R}^\ell$, is

$$v(\mathrm{y}_o) = \inf_{u \in \mathcal{U}_{ad}} \hat{J}^\ell(\mathrm{y}_o, u),$$

where $\hat{J}^\ell(\mathrm{y}_o) = J(y^\ell, u)$, with $y^\ell = \sum_{i=1}^d \mathrm{y}_i^\ell \psi_i$, and y^ℓ solves the dynamical system in (4.13) with initial condition y_o and control input u.

6.3. Numerical Experiments

Numerical experiments are carried out for stabilization problems, i.e., z is chosen to be 0. The parameter settings used for the numerical tests are listed in Table 1. Boundary control is imposed at $x = 0$, i.e., we set $\sigma_0 = \sigma_1 = 0$, and $g = 0$. The left graph of Figure 4 shows the solution of the uncontrolled Burgers equation with the initial condition $y_o = (1-x)\sin(3\pi(x-0.5))$. For solving the Burgers equation, the finite-element method is applied for spatial discretization and

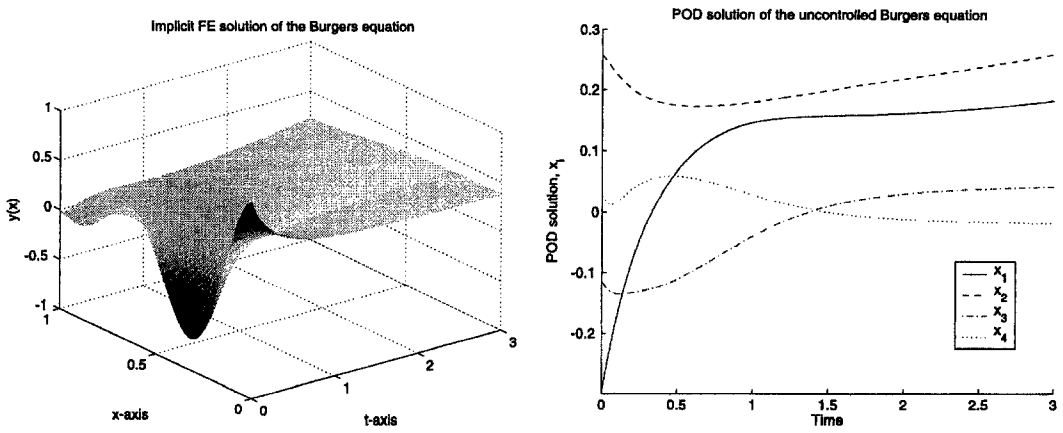

Figure 4. Uncontrolled Burgers solution (left) and its POD solution (right), $\nu = 0.05$.

the implicit Euler method is employed for the time stepping. Further details can be found in [19] and the references cited therein.

Then we take the POD technique described in the previous section to construct the basis functions, four of which are taken to construct the reduced-order model for the Burgers equation. The number of basis functions is chosen such that the ratio $\sum_{i=1}^{l} \lambda_i / \sum_{i=1}^{d} \lambda_i > 98\%$. This ratio is an index indicating how much energy is captured by the first 4 basis functions. In the graph on the right of Figure 4 the coefficients of the POD solution to the uncontrolled reduced-order ODEs are shown. The feedback design is carried out on this resulting four-dimensional ODEs.

Concerning the spatial domain Υ on which the HJB is solved we have to choose it in such a way that it contains the anticipated optimal trajectory $y^*(\cdot)$ of (2.2) for all $t \in [0, T]$. In the case of stabilization a practical procedure consists in considering maxima and minima over the time horizon of the coordinates of the uncontrolled solution, see Figure 4, to add small safety margins and to let this define the polyhedric domain Υ on which the HJB needs to be solved. For our computations we took a uniform $12 \times 8 \times 6 \times 6$ discretization of this polyhedron. We next address the issue of possible boundary conditions.

REMARK 6.1. If the viability assumption is satisfied for the bounded polyhedron $\Upsilon \in \mathbb{R}^4$, i.e., if

$$F(x, u) \cdot n(x) \leq 0, \qquad \forall x \in \bar{\Upsilon}, \quad \forall u \in U,$$

holds with $n(\cdot)$ the exterior normal to $\bar{\Upsilon}$, no boundary condition is needed for the HJB equation. However, in the practical implemention this condition may not be satisfied. In this case we imposed the artificial boundary condition

$$\frac{\partial^2 v}{\partial x^2} = 0, \qquad \forall x \in \bar{\Upsilon},$$

as a remedy.

This time we apply the third-order approximation for the gradients of the value function. We also tested the first-order approximations but the results are unsatisfactory and will therefore not be shown here. Recall that the first-order approximations were sufficient for the 1-D cases in which much finer grid systems could be employed. The time step is set as $\Delta t = 0.002$ to satisfy the CFL condition (3.4).

In numerical computations, the local Lax-Friedrichs flux is constructed independently at each point of the grid system. Therefore, we can apply the so-called fine-grained parallel algorithm [21] to speed up computations. In this parallelism, the same set of codes runs simultaneously on different pieces of data on various processors. First, the required data are transferred to the

slaves. Upon receiving data, the slaves can perform computations, namely constructing the local Lax-Friedrichs flux on the specified grid points, concurrently. After the computations are done on all of the slaves, the data are collected from them. The distribution of the computations to be perfomed on each slave can be determined by the overall performance (size of memory, speed of CPU, etc.) of the slave computers. In this parallel process, the technique of *message passing interface* (MPI) [22] is used to transfer data between the master and slaves.

First the discount rate, μ, is set to 2. In this case, the cost functional without any control is 0.01800. Using optimal control the cost functional is decreased to 0.00675. This value agrees well with the value function 0.00664, and can therefore be used as a first validation of the proposed procedure and its numerical implementation. The results are displayed in the first column of Table 2. In Figure 5 the value function and the associated optimal control are shown as a function of x_1, x_2 plane for constant x_3 and x_4, at the time instance of $t = 3$. The optimal state

Table 1. Parameter settings.

Symbol	Value	Description
ν	0.05	viscosity coefficient
β	0.05	weighting coefficient for the control
T	3	time horizon
z	0	desired state

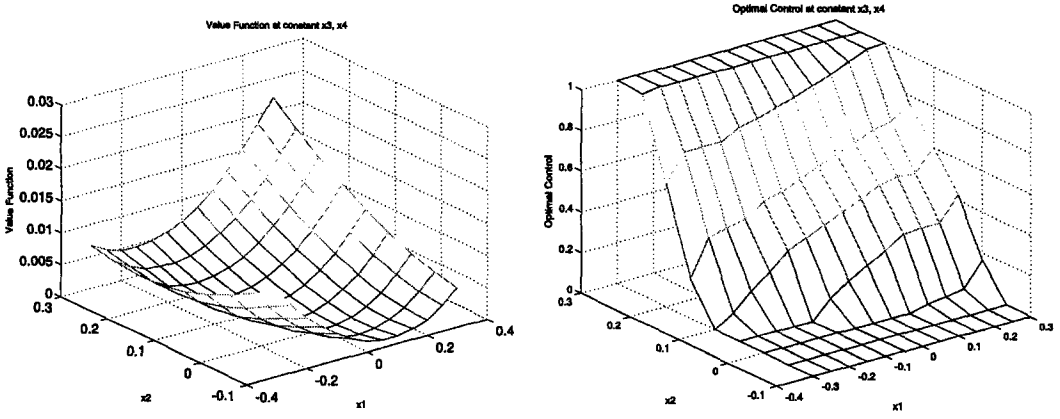

Figure 5. Value function (left) and the corresponding optimal control (right) at constant x_3, x_4, $t = 3$, $\beta = 0.05$, $\mu = 2$.

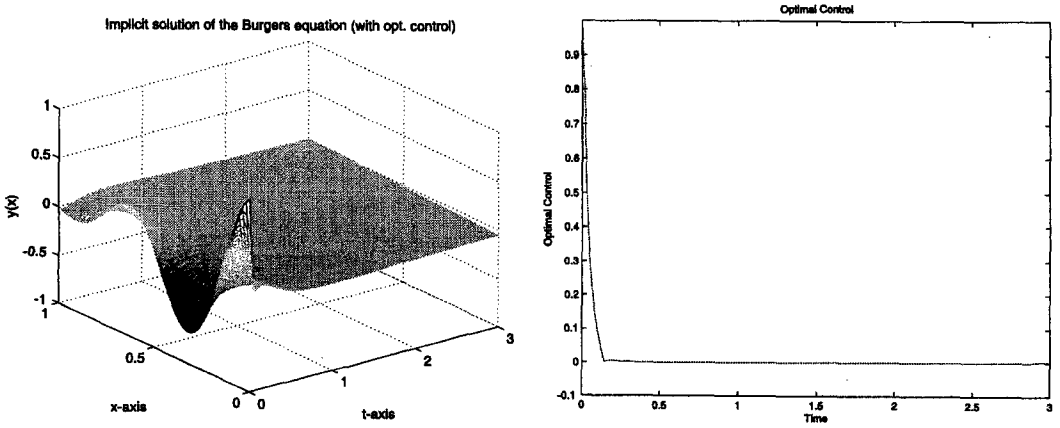

Figure 6. Optimal state (left) and control (right), $\beta = 0.05$, $\mu = 2$.

Table 2. Computational results at $\beta = 0.05$.

	$\mu = 2$	$\mu = 0$
J w Opt. Control	0.00675	0.00886
J w/o Control	0.01800	0.1121
Value Function	0.00664	0.00869

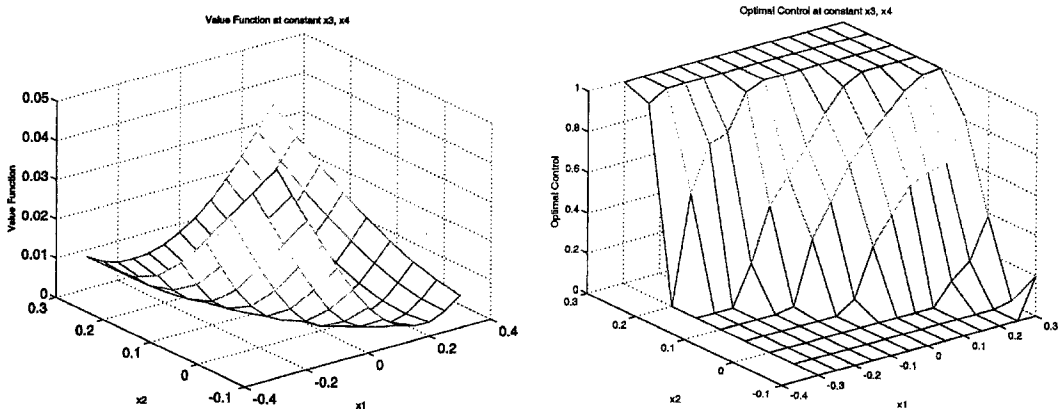

Figure 7. Value function (left) and the corresponding optimal control (right) at constant x_3, x_4, $t = 3$, $\beta = 0.05$, $\mu = 0$.

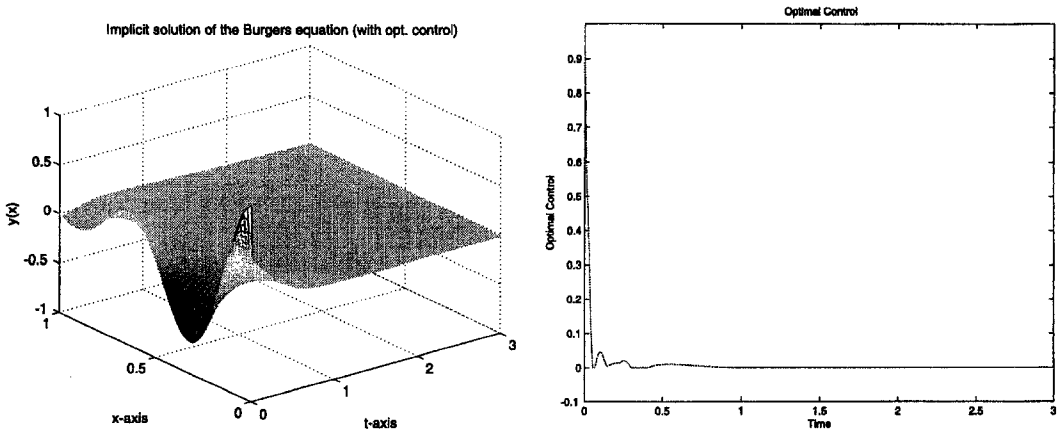

Figure 8. Optimal state (left) and control (right), $\beta = 0.05$, $\mu = 0$.

and the optimal control are plotted in Figure 6. For the purpose of comparison, we also carry out the open-loop design for this case, which gives the optimal cost functional 0.00661. This is slightly less than what we obtained from feedback design. This open-loop design law will be used to demonstrate the effect of noise for open-loop control.

We also tested the HJB equation with discount factor $\mu = 0$. As shown in the second column of Table 2, the cost functional is reduced from 0.1121 to 0.00886. Meanwhile the value function takes the value of 0.00869. Again we observe very good agreement between the optimal cost functional and the value function. The value function and the optimal control at the final time are shown in Figure 7. In Figure 8, the optimal state and the corresponding optimal control for the experiment with $\mu = 0$ are plotted.

Finally we carry out an experiment to demonstrate the effect of noise on the open-loop and the feedback design respectively. The control laws are those computed for $\mu = 2$. The first test is carried out by imposing uniform biased noise in the interval [0, 0.5] in the initial condition. In the second test, we impose uniform biased noise in the interval $[0, 1]$ to the right-hand side

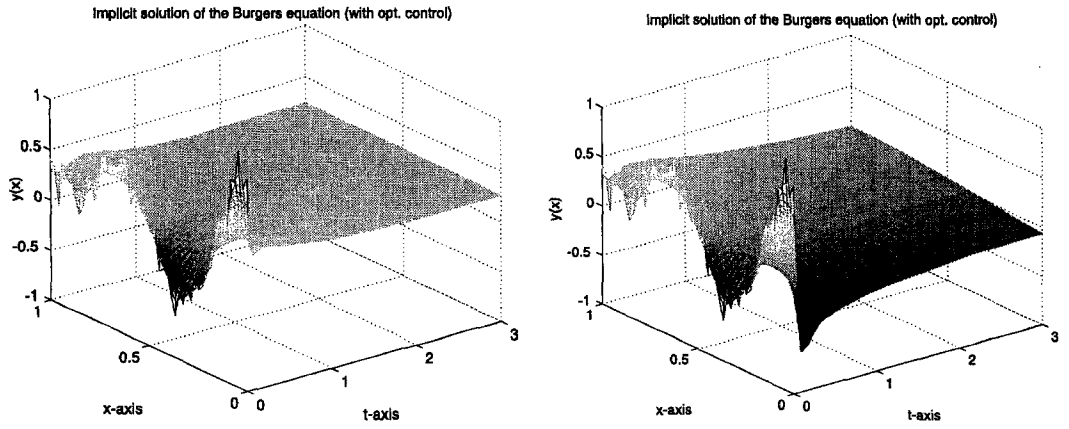

Figure 9.Optimal state: uniform noise of magnitude 0.5 imposed in the initial condition: open-loop design (left) and feedback design (right).

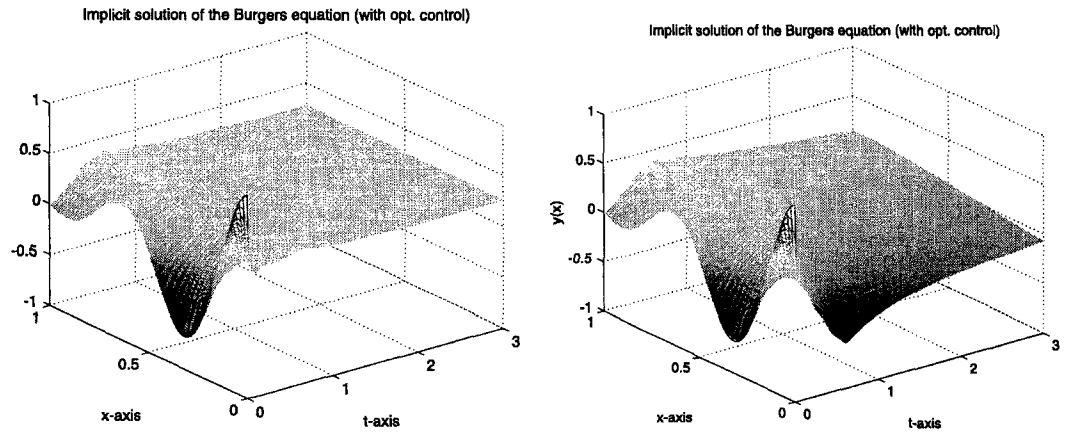

Figure 10. Optimal state with uniform noise of magnitude 1 imposed on the RHS of Burgers equation in a patch of space-time space: open-loop design (left) and feedback desgin (right).

of the Burgers equation in the space-time region $(0,1) \times (0,.6)$. The feedback map can be readily retrieved from the solution of HJB equation. The resulting evolutionary optimal states are presented in Figure 9 and Figure 10. A comparison shows that the open-loop design fails to drive the state to the origin, whereas the closed loop control achieves the design objective.

REFERENCES

1. H.V. Ly and H.T. Tran, Proper orthogonal decomposition for flow calculations and optimal control in a horizontal CVD reactor, *Quart. Appl. Math.* **60** (4), 631–656, (2002).
2. S.C. Beeler, H.T. Tran and H.T. Banks, Feedback control methodologies for nonlinear systems, *J. Optim. Theory Appl.* **107** (1), 1–33, (2000).
3. K. Zhou, J.C. Doyle and K. Glover, *Robust and Optimal Control*, Prentice Hall, New Jersey, (1996).
4. K. Ito and J.D. Schröter, Reduced order feedback synthesis for viscous incompressible flows, *Mathl. Comput. Modelling* **33** (1–3), 173–192, (2001).
5. J.K. Atwell and B.B. King, Proper orthogonal decomposition for reduced basis feedback controllers for parabolic equations, *Mathl. Comput. Modelling* **33** (1–3), 1–19, (2001).
6. H.V. Ly and H.T. Tran, Modelling and control of physical processes using proper orthogonal decomposition, *Mathl. Comput. Modelling* **33** (1–3), 223–236, (2001).
7. K. Kunisch and S. Volkwein, Control of Burgers' equation by a reduced-order approach using Proper Orthogonal decomposition, *J. Optim. Theory and Appl.* **102**, 345–371, (1998).
8. K. Kunisch and S. Volkwein, Galerkin proper orthogonal decomposition methods for a general equation in fluid dynamics, *SIAM J. Numer. Anal.* **40**, 492–515, (2002).

9. C.-S. Huang, S. Wang and K.L. Teo, Solving Hamilton-Jacobi-Bellman equations by a modified method of characteristics, (preprint).
10. S. Wang, F. Gao and K.L. Teo, An upwind finite difference method for the approximation of viscosity solutions to Hamilton-Jacobi-Bellman equations, (preprint).
11. M. Bardi and I. Capuzzo-Dolcetta, *Optimal Control and Viscosity Solutions of Hamilton-Jacobi-Bellman Equations*, Systems & Control: Foundations & Applications, Birkhäuser, (1997).
12. H.-Z. Tang, T. Tang and P. Zhang, An adaptive mesh redistribution method for nonlinear Hamilton-Jacobi equations in two- and three-dimensions, (preprint).
13. S. Gottlieb and C. -W. Shu, Total variation diminishing Runge-Kutta schemes, ICASE Report 96-50, (1996).
14. G.-S. Jiang and D. Peng, Weighted ENO schemes for Hamilton-Jacobi equations, *SIAM J. Scientific Computing* **21**, 2126–2143, (2000).
15. G. Berkooz, P. Holmes, and J.L. Lumley, *Turbulence, Coherent Structures, Dynamical Systems and Symmetry, Cambridge Monographs on Mechanics*, Cambridge University Press, (1996).
16. R. Dautray and J.-L. Lions, *Mathematical Analysis and Numerical Methods for Science and Technology, Volume 5: Evolution Problems I*, Springer-Verlag, Berlin, (1992).
17. R. Temam, *Infinite-Dimensional Dynamical Systems in Mechanics and Physics, Volume 68 of Applied Mathematical Sciences*, Springer-Verlag, New York, (1988).
18. A. Kurganov, S. Noelle and G. Petrova, Semidiscrete central-upwind schemes for hyperbolic conservation laws and Hamilton-Jacobi equations, *SIAM J. Sci. Comput.* **23** (3), 707–740, (2001).
19. S. Volkwein, Lagrange-SQP techniques for the control constrained optimal control problems for the Burgers equation, *Computational Optimization and Applications* **26**, 253–284, (2003).
20. S. Volkwein, Second-order conditions for boundary control problem of the Burgers equation, *Control and Cybernetics* **30**, 249–278, (2001).
21. T. Furlani, Introduction to Parallel Computing at CCR, `http://www.ccr.buffalo.edu/documents/CCR-parallel_process_intro.pdf`.
22. E. Heiberg, MATLAB Parallization Toolkit, `http://hem.passagen.se/einar_heiberg/`.