



A GRADIENT-BASED KERNEL OPTIMIZATION APPROACH FOR PARABOLIC DISTRIBUTED PARAMETER CONTROL SYSTEMS

ZHIGANG REN, CHAO XU* QUN LIN AND RYAN LOXTON

Abstract: This paper proposes a new gradient-based optimization approach for designing optimal feedback kernels for parabolic distributed parameter systems with boundary control. Unlike traditional kernel optimization methods for parabolic systems, our new method does not require solving non-standard Riccati-type or Klein-Gorden-type partial differential equations (PDEs). Instead, the feedback kernel is parameterized as a second-order polynomial whose coefficients are decision variables to be tuned via gradient-based dynamic optimization, where the gradient of the system cost functional (which penalizes both kernel and output magnitude) is computed by solving a so-called “costate” PDE in standard form. Special constraints are imposed on the kernel coefficients to ensure that, under mild conditions, the optimized kernel yields closed-loop stability. Numerical simulations demonstrate the effectiveness of the proposed approach.

Key words: *optimization, feedback kernel design, boundary stabilization, parabolic PDE systems.*

Mathematics Subject Classification: *49M37, 90C30, 93C20.*

1 Introduction

Spatial-temporal evolutionary processes (STEPS) are usually modeled by partial differential equations (PDEs), which are commonly referred to as distributed parameter systems (DPS) in the field of automatic control. The parabolic system is an important type of DPS that describes a wide range of natural phenomena, including diffusion, heat transfer, and fusion plasma transport. Over the past few decades, control theory for the parabolic DPS has developed into a mature research topic at the interface of engineering and applied mathematics [2, 4, 14].

There are two main control synthesis approaches for the parabolic DPS: the *design-then-discretize* framework and the *discretize-then-design* framework. In the design-then-discretize framework, analytical techniques are first applied to derive equations for the optimal controller—e.g., the Riccati equation for optimal control or the backstepping kernel equation for boundary stabilization—and then these equations are solved using numerical discretization techniques. In the discretize-then-design framework, the sequence is reversed: the infinite dimensional PDE system is first discretized to obtain a finite dimensional system, and then various controller synthesis and numerical optimization techniques are applied to

*Corresponding author. This work was supported by the National Science Foundation of China (grant 61473253), the National High Technology Research and Development Program of China (grant 2012AA041701), and the Collaborative Innovation Center for Industrial Cyber-Physical Systems.

solve the corresponding discretized problem. Popular approaches for obtaining the finite dimensional approximation in the discretize-then-design framework include mesh-associated discretization techniques and model order reduction (MOR) techniques. Examples of mesh-associated discretization techniques include the finite difference method [17], the finite element method [17], the finite volume method [17], and the spectral method [5]. MOR methods include the proper orthogonal decomposition method [6] and the balanced truncation method [15], both of which exploit system input-output properties [1]. Since MOR techniques can generate low-order models without compromising solution accuracy, they are popular for dealing with complex STEPs, which arise frequently in applications such as plasma physics, fluid flow, and heat and mass transfer (e.g., see [24] and the references cited therein).

The linear quadratic (LQ) control framework, a widely-used technique in controller synthesis, is well-defined in infinite dimensional function spaces to deal with the parabolic DPS (e.g., [2, 4]). However, the LQ control framework requires solving Riccati-type differential equations, which are nonlinear parabolic PDEs of dimension one greater than the original parabolic PDE system. For example, to generate an optimal feedback controller for a scalar heat equation, a Riccati PDE defined over a rectangular domain must be solved [16]. Hence, the LQ approach does not actually solve the controller synthesis problem directly, but instead converts it into another problem (i.e., solve a Riccati-type PDE) that is still extremely difficult to solve from a computational point of view.

One of the major advances in PDE control in recent years has been the so-called infinite dimensional Volterra integral feedback, or the backstepping method (e.g., [9, 13]). Instead of Riccati-type PDEs, the backstepping method requires solving the so-called kernel equations—linear Klein-Gorden-type PDEs for which the successive approach can be used to obtain explicit solutions. This method was originally developed for the stabilization of one dimensional parabolic DPS and then extended to fluid flows [21], magnetohydrodynamic flows [22, 25], and elastic vibration [8]. In addition, the backstepping method can also be applied to achieve full state feedback stabilization and state estimation of PDE-ODE cascade systems [18].

In this paper, we propose a new framework for control synthesis for the parabolic DPS. This new framework does not require solving Riccati-type or Klein-Gorden-type PDEs. Instead, it requires solving a so-called “costate” PDE, which is much easier to solve from a computational viewpoint. In fact, many numerical software packages, such as Comsol Multiphysics and MATLAB PDE ToolBox, can be used to generate numerical solutions for the costate PDE. The Riccati PDEs, on the other hand, are usually not in standard form and thus cannot be solved using off-the-shelf software packages. The optimization approach proposed in this paper is motivated by the well-known PID tuning problem, in which the coefficients in a PID controller need to be selected judiciously to optimize system performance. Relevant literature includes reference [7], where extremum seeking algorithms are used to tune the PID parameters; reference [10], where the PID tuning problem is reformulated into a nonlinear optimization problem, and subsequently solved using numerical optimization techniques; and reference [23], where the iterative learning tuning method is used to update the PID parameters whenever a control task is repeated. The current paper can be viewed as an extension of these optimization-based feedback design ideas to infinite dimensional systems.

The remainder of this paper is organized as follows. In Section 2, we formulate two parameter optimization problems for a class of unstable linear parabolic diffusion-reaction PDEs with control actuation at the boundary: the first problem involves optimizing a set of parameters that govern the feedback kernel; the second problem is a modification of the first

problem with additional constraints to ensure closed-loop stability. In Section 3, we derive the gradients of the cost and constraint functions for the optimization problems in Section 2. Then, in Section 4, we present a numerical algorithm, which is based on the results obtained in Section 3, for determining the optimal feedback kernel. Section 5 presents the numerical simulation results. Finally, Section 6 concludes the paper by proposing some further research topics.

2 Problem Formulation

2.1 Feedback Kernel Optimization

We consider the following parabolic PDE system:

$$\begin{cases} y_t(x, t) = y_{xx}(x, t) + cy(x, t), & (2.1a) \\ y(0, t) = 0, & (2.1b) \\ y(1, t) = u(t), & (2.1c) \\ y(x, 0) = y_0(x), & (2.1d) \end{cases}$$

where $c > 0$ is a given constant and $u(t)$ is a boundary control. It is well known that the uncontrolled version of system (2.1) is unstable when the constant c is sufficiently large [9]. Thus, it is necessary to design an appropriate feedback control law for $u(t)$ to stabilize the system. According to the LQ control [16] and backstepping synthesis approaches [9], the optimal feedback control law takes the following form:

$$u(t) = \int_0^1 \mathcal{K}(1, \xi)y(\xi, t)d\xi, \tag{2.2}$$

where the feedback kernel $\mathcal{K}(1, \xi)$ is obtained by solving either a Riccati-type or a Klein-Gorden-type PDE. By introducing the new notation $k(\xi) = \mathcal{K}(1, \xi)$, we can write the feedback control policy (2.2) in the following form:

$$u(t) = \int_0^1 k(\xi)y(\xi, t)d\xi.$$

The corresponding closed-loop system is

$$\begin{cases} y_t(x, t) = y_{xx}(x, t) + cy(x, t), & (2.3a) \\ y(x, 0) = y_0(x), & (2.3b) \\ y(0, t) = 0, & (2.3c) \\ y(1, t) = \int_0^1 k(\xi)y(\xi, t)d\xi. & (2.3d) \end{cases}$$

In reference [9], the backstepping method is used to express the optimal feedback kernel in terms of the first-order modified Bessel function. More specifically,

$$\mathcal{K}(1, \xi) = -c\xi \frac{I_1(\sqrt{c(1-\xi^2)})}{\sqrt{c(1-\xi^2)}}, \tag{2.4}$$

where I_1 is the first-order modified Bessel function given by

$$I_1(\omega) = \sum_{n=0}^{\infty} \frac{\omega^{2n+1}}{2^{2n+1}n!(n+1)!}.$$

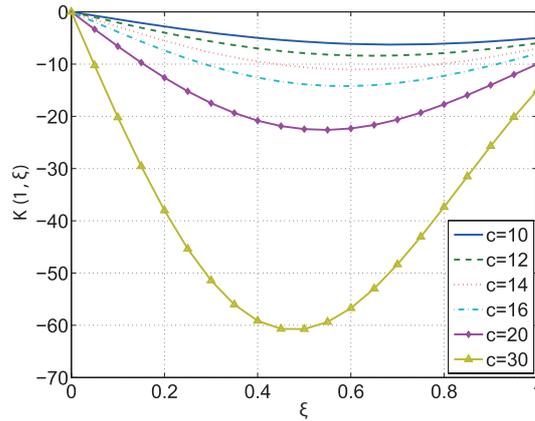


Figure 1: The feedback kernel (2.4) for various values of c .

The feedback kernel (2.4) is plotted in Figure 1 for different values of c . Note that its shape is similar to a quadratic function. Note also that $\mathcal{K}(1, \xi) = 0$ when $\xi = 0$. Accordingly, motivated by the quadratic behavior exhibited in Figure 1, we express $k(\xi)$ in the following parameterized form:

$$k(\xi; \Theta) = \theta_1 \xi + \theta_2 \xi^2, \tag{2.5}$$

where $\Theta = (\theta_1, \theta_2)^\top$ is a parameter vector to be optimized.

Moreover, we assume that the parameters must satisfy the following bound constraints:

$$a_1 \leq \theta_1 \leq b_1, \quad a_2 \leq \theta_2 \leq b_2, \tag{2.6}$$

where a_1, a_2, b_1 and b_2 are given bounds.

Let $y(x, t; \Theta)$ denote the solution of the closed-loop system (2.3) with the parameterized kernel (2.5). The results in [20] ensure that such a solution exists and is unique.

Our goal is to stabilize the closed-loop system with minimal energy input. Accordingly, we consider the following cost functional:

$$g_0(\Theta) = \frac{1}{2} \int_0^T \int_0^1 y^2(x, t; \Theta) dx dt + \frac{1}{2} \int_0^1 k^2(x; \Theta) dx. \tag{2.7}$$

This cost functional consists of two terms: the first term penalizes output deviation from zero (stabilization); the second term penalizes kernel magnitude (energy minimization). We now state our kernel optimization problem formally as follows.

Problem P₁. Given the PDE system (2.3) with the parameterized kernel (2.5), find an optimal parameter vector $\Theta = (\theta_1, \theta_2)^\top$ such that the cost functional (2.7) is minimized subject to the bound constraints (2.6).

2.2 Closed-Loop Stability

Since (2.7) is a finite-time cost functional, there is no guarantee that the optimized kernel (2.5) generated by the solution of Problem P₁ stabilizes the closed-loop system (2.3) as $t \rightarrow \infty$. Nevertheless, we now show that, by analyzing the solution structure of (2.3), additional constraints can be added to Problem P₁ to ensure closed-loop stability.

Using the separation of variables approach, we decompose $y(x, t)$ as follows:

$$y(x, t) = \mathcal{X}(x)\mathcal{T}(t). \tag{2.8}$$

Substituting (2.8) into (2.3a), we obtain

$$\mathcal{X}(x)\dot{\mathcal{T}}(t) = \mathcal{X}''(x)\mathcal{T}(t) + c\mathcal{X}(x)\mathcal{T}(t), \tag{2.9}$$

where

$$\begin{aligned} \dot{\mathcal{T}}(t) &= \frac{d\mathcal{T}(t)}{dt}, \\ \mathcal{X}''(x) &= \frac{d^2\mathcal{X}(x)}{dx^2}. \end{aligned}$$

Furthermore, from the boundary conditions (2.3c) and (2.3d),

$$\mathcal{X}(0)\mathcal{T}(t) = 0, \quad \mathcal{X}(1)\mathcal{T}(t) = \int_0^1 k(\xi; \Theta)\mathcal{X}(\xi)\mathcal{T}(t)d\xi.$$

Thus, we immediately obtain

$$\mathcal{X}(0) = 0, \tag{2.10}$$

$$\mathcal{X}(1) = \int_0^1 k(\xi; \Theta)\mathcal{X}(\xi)d\xi. \tag{2.11}$$

Rearranging (2.9) gives

$$\frac{\mathcal{X}''(x) + c\mathcal{X}(x)}{\mathcal{X}(x)} = \frac{\dot{\mathcal{T}}(t)}{\mathcal{T}(t)}.$$

This equation must hold for all x and t . Hence, there exists a constant σ (an eigenvalue) such that

$$\frac{\mathcal{X}''(x) + c\mathcal{X}(x)}{\mathcal{X}(x)} = \frac{\dot{\mathcal{T}}(t)}{\mathcal{T}(t)} = \sigma. \tag{2.12}$$

Clearly,

$$\mathcal{T}(t) = T_0e^{\sigma t}, \tag{2.13}$$

where $T_0 = \mathcal{T}(0)$ is a constant to be determined.

To solve for $\mathcal{X}(x)$, we must consider three cases: (i) $c < \sigma$; (ii) $c = \sigma$; (iii) $c > \sigma$. In cases (i) and (ii), the general solutions of (2.12) are, respectively,

$$\mathcal{X}(x) = X_0e^{\sqrt{\sigma-c}x} + X_1e^{-\sqrt{\sigma-c}x},$$

and

$$\mathcal{X}(x) = X_0 + X_1x,$$

where X_0 and X_1 are constants to be determined from the boundary conditions (2.10) and (2.11). Then the corresponding solutions of (2.3) are

$$y(x, t) = X_0T_0e^{\sqrt{\sigma-c}x+\sigma t} + X_1T_0e^{-\sqrt{\sigma-c}x+\sigma t},$$

and

$$y(x, t) = X_0T_0e^{\sigma t} + X_1T_0xe^{\sigma t}.$$

These solutions are clearly unstable because $0 < c \leq \sigma$. Thus, we want to choose the parameters θ_1 and θ_2 so that the unique solution of (2.3) satisfies case (iii) instead of cases (i) and (ii).

In case (iii), the general solution of (2.12) is

$$\mathcal{X}(x) = X_0 \cos(\sqrt{c-\sigma}x) + X_1 \sin(\sqrt{c-\sigma}x), \quad (2.14)$$

where X_0 and X_1 are constants to be determined from the boundary conditions (2.10) and (2.11). Substituting (2.14) into (2.10), we obtain

$$\mathcal{X}(0) = X_0 = 0.$$

Hence,

$$\mathcal{X}(x) = X_1 \sin(\sqrt{c-\sigma}x). \quad (2.15)$$

To simplify the notation, we introduce a new variable $\alpha = \sqrt{c-\sigma}$. Substituting (2.15) into condition (2.11), we have

$$X_1 \sin \alpha = X_1 \int_0^1 \theta_1 \xi \sin(\alpha \xi) d\xi + X_1 \int_0^1 \theta_2 \xi^2 \sin(\alpha \xi) d\xi,$$

and thus

$$\sin \alpha = \int_0^1 \theta_1 \xi \sin(\alpha \xi) d\xi + \int_0^1 \theta_2 \xi^2 \sin(\alpha \xi) d\xi. \quad (2.16)$$

Evaluating the first integral on the right hand side of (2.16) gives

$$\int_0^1 \theta_1 \xi \sin(\alpha \xi) d\xi = \theta_1 \left(\frac{\sin \alpha}{\alpha^2} - \frac{\cos \alpha}{\alpha} \right). \quad (2.17)$$

Evaluating the second integral on the right hand side of (2.16) gives

$$\int_0^1 \theta_2 \xi^2 \sin(\alpha \xi) d\xi = -\theta_2 \left(\frac{\cos \alpha}{\alpha} - \frac{2 \sin \alpha}{\alpha^2} \right) + \frac{2\theta_2(\cos \alpha - 1)}{\alpha^3}. \quad (2.18)$$

Thus, using (2.17) and (2.18), (2.16) can be simplified as

$$(\theta_1 \alpha^2 + \theta_2 \alpha^2 - 2\theta_2) \cos \alpha + (\alpha^3 - \theta_1 \alpha - 2\theta_2 \alpha) \sin \alpha + 2\theta_2 = 0. \quad (2.19)$$

The following result, the proof of which is deferred to the appendix, is fundamental to our subsequent analysis.

Lemma 2.1. *Suppose $\Theta = (\theta_1, \theta_2)^\top$ satisfies the following inequality:*

$$\theta_1^2 + \theta_2^2 + 2\theta_1\theta_2 - 2\theta_1 - 4\theta_2 \geq 0. \quad (2.20)$$

Then equation (2.19) has an infinite number of positive solutions.

For any α satisfying (2.19), there exists a corresponding solution of (2.12) in the form (2.15). Let $\{\alpha_n\}_{n=1}^\infty$ be a sequence of positive solutions of (2.19). Then the general solution of (2.12) is

$$\mathcal{X}(x) = \sum_{n=1}^{\infty} A_n \sin(\alpha_n x),$$

where A_n are constants to be determined. The corresponding eigenvalues are

$$\sigma_n = c - \alpha_n^2, \quad n = 1, 2, 3, \dots$$

Hence, using (2.13),

$$y(x, t) = \sum_{n=1}^{\infty} T_0 A_n e^{(c-\alpha_n^2)t} \sin(\alpha_n x). \tag{2.21}$$

By virtue of (2.10) and (2.11), this solution satisfies the boundary conditions (2.3c) and (2.3d). The constants T_0 and A_n must be selected appropriately so that the initial condition (2.3b) is also satisfied. To ensure stability as $t \rightarrow \infty$, each eigenvalue $\sigma_n = c - \alpha_n^2$ in (2.21) must be negative. Thus, we impose the following constraints on $\Theta = (\theta_1, \theta_2)^\top$:

$$\theta_1^2 + \theta_2^2 + 2\theta_1\theta_2 - 2\theta_1 - 4\theta_2 \geq 0, \tag{2.22a}$$

$$c - \alpha^2 \leq -\epsilon, \tag{2.22b}$$

$$(\theta_1\alpha^2 + \theta_2\alpha^2 - 2\theta_2) \cos \alpha + (\alpha^3 - \theta_1\alpha - 2\theta_2\alpha) \sin \alpha + 2\theta_2 = 0, \tag{2.22c}$$

where ϵ is a given positive parameter and α is the smallest positive solution of (2.19). Note that α here is treated as an additional optimization variable. Constraint (2.22a) ensures that there are an infinite number of eigenvalues (see Lemma 2.1) and thus the solution form (2.21) is valid. Constraints (2.22b) and (2.22c) ensure that the largest eigenvalue is negative, thus guaranteeing solution stability. Adding constraints (2.22) to Problem P_1 yields the following modified problem.

Problem P_2 . Given the PDE system (2.3) with the parameterized kernel (2.5), choose $\Theta = (\theta_1, \theta_2)^\top$ and α such that the cost functional (2.7) is minimized subject to the bound constraints (2.6) and the nonlinear constraints (2.22).

The next result is concerned with the stability of the closed-loop system corresponding to the optimized kernel from Problem P_2 .

Theorem 2.2. *Let (Θ^*, α^*) be an optimal solution of Problem P_2 , where α^* is the smallest positive solution of equation (2.22c) corresponding to Θ^* . Suppose that there exists a sequence $\{\alpha_n^*\}_{n=1}^{\infty}$ of positive solutions to equation (2.22c) corresponding to Θ^* such that $y_0(x) \in \text{span}\{\sin(\alpha_n^* x)\}$. Then the closed-loop system (2.3) corresponding to Θ^* is stable.*

Proof. Because of constraint (2.22a), the solution form (2.21) with $\alpha_n = \alpha_n^*$ is guaranteed to satisfy (2.3a), (2.3c) and (2.3d). If $y_0(x) \in \text{span}\{\sin(\alpha_n^* x)\}$, then there exists constants $Y_n, n \geq 1$, such that

$$y_0(x) = \sum_{n=1}^{\infty} Y_n \sin(\alpha_n^* x).$$

Taking $Y_n = T_0 A_n$ ensures that (2.21) with $\alpha_n = \alpha_n^*$ also satisfies the initial conditions (2.3b), and is therefore the unique solution of (2.3). Since α^* is the first positive solution of equation (2.22c), it follows from constraint (2.22b) that for each $n \geq 1$,

$$c - (\alpha_n^*)^2 \leq c - (\alpha^*)^2 \leq -\epsilon < 0.$$

This shows that all eigenvalues are negative. □

Theorem 2.2 requires that the initial function $y_0(x)$ be contained within the linear span of sinusoidal functions $\sin(\alpha_n^* x)$, where each α_n^* is a solution of equation (2.19) corresponding to Θ^* . The good thing about this condition is that it can be verified numerically by solving the following optimization problem: choose span coefficients $Y_n, 1 \leq n \leq N$, to minimize

$$J = \int_0^1 \left| y_0(x) - \sum_{n=1}^N Y_n \sin(\alpha_n^* x) \right|^2 dx, \tag{2.23}$$

where N is a sufficiently large integer and each α_n^* is a solution of equation (2.19) corresponding to the optimal solution of Problem P₂. If the optimal cost value for this optimization problem is sufficiently small, then the span condition in Theorem 2.2 is likely to be satisfied, and therefore closed-loop stability is guaranteed.

Based on our computational experience, the span condition in Theorem 2.2 is usually satisfied. This can be explained as follows. In the proof of Lemma 2.1 (see the appendix), we show that for any $\epsilon \in (0, \frac{1}{2}\pi)$, there exists at least one solution of (2.19) in the interval $[k\pi - \epsilon, k\pi + \epsilon]$ when k is sufficiently large. It follows that $k\pi$ is an approximate solution of (2.19) for all sufficiently large k —in a sense, the solutions α_n^* of (2.19) converge to the integer multiples of π . In our computational experience, this convergence occurs very rapidly. Thus, it is reasonable to expect that the linear span of $\{\sin(\alpha_n^*x)\}$ is “approximately” the same as the linear span of $\{\sin(n\pi x)\}$, which is known to be a basis for the space of continuous functions defined on $[0, 1]$.

3 Gradient Computation

Problem P₂ is an optimal parameter selection problem with decision parameters θ_1, θ_2 and α . In principle, such problems can be solved as nonlinear optimization problems using the Sequential Quadratic Programming (SQP) method or other nonlinear optimization methods. However, to do this, we need the gradients of the cost functional (2.7) and the constraint functions (2.22) with respect to the decision parameters. The gradients of the constraint functions can be easily derived using elementary differentiation. Define

$$\begin{aligned} g_1(\Theta) &= \theta_1^2 + \theta_2^2 + 2\theta_1\theta_2 - 2\theta_1 - 4\theta_2, \\ g_2(\alpha) &= c - \alpha^2, \\ g_3(\Theta, \alpha) &= (\theta_1\alpha^2 + \theta_2\alpha^2 - 2\theta_2)\cos\alpha + (\alpha^3 - \theta_1\alpha - 2\theta_2\alpha)\sin\alpha + 2\theta_2. \end{aligned}$$

Then the corresponding constraint gradients are given by

$$\nabla_{\theta_1}g_1(\Theta) = 2\theta_1 + 2\theta_2 - 2, \quad \nabla_{\theta_2}g_1(\Theta) = 2\theta_2 + 2\theta_1 - 4, \quad \nabla_{\alpha}g_2(\alpha) = -2\alpha, \quad (3.1)$$

and

$$\nabla_{\theta_1}g_3(\Theta, \alpha) = \alpha^2 \cos\alpha - \alpha \sin\alpha, \quad (3.2a)$$

$$\nabla_{\theta_2}g_3(\Theta, \alpha) = (\alpha^2 - 2)\cos\alpha - 2\alpha \sin\alpha + 2, \quad (3.2b)$$

$$\nabla_{\alpha}g_3(\Theta, \alpha) = (\alpha^3 + \theta_1\alpha)\cos\alpha + (3\alpha^2 - \theta_1\alpha^2 - \theta_2\alpha^2 - \theta_1)\sin\alpha. \quad (3.2c)$$

Since the constraint functions in (2.22) are explicit functions of the decision variables, their gradients are easily obtained. The cost functional (2.7), on the other hand, is an implicit function of Θ because it depends on the state trajectory $y(x, t)$. Thus, computing the gradient of (2.7) is a non-trivial task. We now develop a computational method, analogous to the costate method in the optimal control of ordinary differential equations [11, 12, 19], for computing this gradient.

We define the following costate PDE system:

$$\begin{cases} v_t(x, t) + v_{xx}(x, t) + cv(x, t) + y(x, t; \Theta) - k(x; \Theta)v_x(1, t) = 0, & (3.3a) \end{cases}$$

$$\begin{cases} v(0, t) = v(1, t) = 0, & (3.3b) \end{cases}$$

$$\begin{cases} v(x, T) = 0. & (3.3c) \end{cases}$$

Let $v(x, t; \Theta)$ denote the solution of the costate PDE system (3.3) corresponding to the parameter vector Θ . Then we have the following theorem.

Theorem 3.1. *The gradient of the cost functional (2.7) is given by*

$$\nabla_{\theta_1} g_0(\Theta) = - \int_0^T \int_0^1 x v_x(1, t; \Theta) y(x, t; \Theta) dx dt + \frac{1}{3} \theta_1 + \frac{1}{4} \theta_2, \quad (3.4a)$$

$$\nabla_{\theta_2} g_0(\Theta) = - \int_0^T \int_0^1 x^2 v_x(1, t; \Theta) y(x, t; \Theta) dx dt + \frac{1}{4} \theta_1 + \frac{1}{5} \theta_2. \quad (3.4b)$$

Proof. Let $\nu(x, t)$ be an arbitrary function satisfying

$$\nu(x, T) = 0, \quad \nu(0, t) = \nu(1, t) = 0. \quad (3.5)$$

Then we can rewrite the cost functional (2.7) in augmented form as follows:

$$\begin{aligned} g_0(\Theta) &= \frac{1}{2} \int_0^T \int_0^1 y^2(x, t; \Theta) dx dt + \frac{1}{2} \int_0^1 k^2(x; \Theta) dx \\ &\quad + \int_0^T \int_0^1 \nu(x, t) \{ -y_t(x, t; \Theta) + y_{xx}(x, t; \Theta) + cy(x, t; \Theta) \} dx dt. \end{aligned} \quad (3.6)$$

Using integration by parts and applying the boundary condition (2.3c), we can simplify the augmented cost functional (3.6) to obtain

$$\begin{aligned} g_0(\Theta) &= \frac{1}{2} \int_0^T \int_0^1 y^2(x, t; \Theta) dx dt + \frac{1}{2} \int_0^1 k^2(x; \Theta) dx \\ &\quad - \int_0^1 \nu(x, T) y(x, T; \Theta) dx + \int_0^1 \nu(x, 0) y(x, 0) dx \\ &\quad + \int_0^T \int_0^1 \nu_t(x, t) y(x, t; \Theta) dx dt + \int_0^T [\nu(x, t) y_x(x, t; \Theta)]_{x=0}^{x=1} dt \\ &\quad - \int_0^T \nu_x(1, t) y(1, t; \Theta) dt + \int_0^T \int_0^1 \nu_{xx}(x, t) y(x, t; \Theta) dx dt \\ &\quad + c \int_0^T \int_0^1 \nu(x, t) y(x, t; \Theta) dx dt. \end{aligned}$$

Thus, recalling the conditions (2.3b) and (3.5), we obtain

$$\begin{aligned} g_0(\Theta) &= \frac{1}{2} \int_0^T \int_0^1 y^2(x, t; \Theta) dx dt + \frac{1}{2} \int_0^1 k^2(x; \Theta) dx \\ &\quad + \int_0^T \int_0^1 \{ \nu_t(x, t) + \nu_{xx}(x, t) + c\nu(x, t) \} y(x, t; \Theta) dx dt \\ &\quad + \int_0^1 \nu(x, 0) y_0(x) dx - \int_0^T \nu_x(1, t) y(1, t; \Theta) dt. \end{aligned}$$

Now, consider a perturbation $\varepsilon \rho$ in the parameter vector Θ , where ε is a constant of sufficiently small magnitude and ρ is an arbitrary vector. The corresponding perturbation in the state is,

$$y(x, t; \Theta + \varepsilon \rho) = y(x, t; \Theta) + \varepsilon \langle \nabla_{\Theta} y(x, t; \Theta), \rho \rangle + \mathcal{O}(\varepsilon^2), \quad (3.7)$$

and the perturbation in the feedback kernel is,

$$k(x; \Theta + \varepsilon \rho) = k(x; \Theta) + \varepsilon \langle \nabla_{\Theta} k(x; \Theta), \rho \rangle + \mathcal{O}(\varepsilon^2), \quad (3.8)$$

where $\mathcal{O}(\varepsilon^2)$ denotes omitted second-order terms such that $\varepsilon^{-1}\mathcal{O}(\varepsilon^2) \rightarrow 0$ as $\varepsilon \rightarrow 0$. For notational simplicity, we define $\eta(x, t) = \langle \nabla_{\Theta} y(x, t; \Theta), \rho \rangle$. Obviously, $\eta(x, 0) = 0$, because the initial profile $y_0(x)$ is independent of the parameter vector Θ . Based on (3.7) and (3.8), the perturbed augmented cost functional takes the following form:

$$\begin{aligned} g_0(\Theta + \varepsilon\rho) &= \frac{1}{2} \int_0^T \int_0^1 \{y(x, t; \Theta) + \varepsilon\eta(x, t)\}^2 dx dt \\ &\quad + \int_0^T \int_0^1 \{\nu_t(x, t) + \nu_{xx}(x, t) + c\nu(x, t)\} \{y(x, t; \Theta) + \varepsilon\eta(x, t)\} dx dt \\ &\quad + \int_0^1 \nu(x, 0)y_0(x) dx - \int_0^T \nu_x(1, t) \{y(1, t; \Theta) + \varepsilon\eta(1, t)\} dt \\ &\quad + \frac{1}{2} \int_0^1 \{k(x; \Theta) + \varepsilon \langle \nabla_{\Theta} k(x; \Theta), \rho \rangle\}^2 dx + \mathcal{O}(\varepsilon^2). \end{aligned} \quad (3.9)$$

From the boundary condition in (2.3d), we have

$$\begin{aligned} y(1, t; \Theta) + \varepsilon\eta(1, t) &= \int_0^1 k(x; \Theta) \{y(x, t; \Theta) + \varepsilon\eta(x, t)\} dx \\ &\quad + \int_0^1 \varepsilon \langle \nabla_{\Theta} k(x; \Theta), \rho \rangle y(x, t; \Theta) dx + \mathcal{O}(\varepsilon^2). \end{aligned} \quad (3.10)$$

Substituting (3.10) into (3.9) gives

$$\begin{aligned} g_0(\Theta + \varepsilon\rho) &= \frac{1}{2} \int_0^T \int_0^1 \{y(x, t; \Theta) + \varepsilon\eta(x, t)\}^2 dx dt \\ &\quad + \int_0^T \int_0^1 \{\nu_t(x, t) + \nu_{xx}(x, t) + c\nu(x, t)\} \{y(x, t; \Theta) + \varepsilon\eta(x, t)\} dx dt \\ &\quad + \int_0^1 \nu(x, 0)y_0(x) dx - \int_0^T \nu_x(1, t) \left[\int_0^1 k(x; \Theta) \{y(x, t; \Theta) + \varepsilon\eta(x, t)\} dx \right] dt \\ &\quad - \int_0^T \nu_x(1, t) \left[\int_0^1 \varepsilon \langle \nabla_{\Theta} k(x; \Theta), \rho \rangle y(x, t; \Theta) dx \right] dt \\ &\quad + \frac{1}{2} \int_0^1 \{k(x; \Theta) + \varepsilon \langle \nabla_{\Theta} k(x; \Theta), \rho \rangle\}^2 dx + \mathcal{O}(\varepsilon^2). \end{aligned} \quad (3.11)$$

Taking the derivative of (3.11) with respect to ε and setting $\varepsilon = 0$ gives

$$\begin{aligned} \langle \nabla_{\Theta} g_0(\Theta), \rho \rangle &= \left. \frac{dg_0(\Theta + \varepsilon\rho)}{d\varepsilon} \right|_{\varepsilon=0} \\ &= \int_0^T \int_0^1 \{y(x, t; \Theta) + \nu_t(x, t) + \nu_{xx}(x, t) + c\nu(x, t)\} \eta(x, t) dx dt \\ &\quad - \int_0^T \int_0^1 \nu_x(1, t) k(x; \Theta) \eta(x, t) dx dt \\ &\quad - \int_0^T \int_0^1 \nu_x(1, t) \langle \nabla_{\Theta} k(x; \Theta), \rho \rangle y(x, t; \Theta) dx dt \\ &\quad + \int_0^1 k(x; \Theta) \langle \nabla_{\Theta} k(x; \Theta), \rho \rangle dx. \end{aligned} \quad (3.12)$$

Choosing the multiplier $\nu(x, t)$ to be the solution of the costate system (3.3), the gradient in (3.12) becomes

$$\begin{aligned} \langle \nabla_{\Theta} g_0(\Theta), \rho \rangle &= - \int_0^T \int_0^1 v_x(1, t; \Theta) \langle \nabla_{\Theta} k(x; \Theta), \rho \rangle y(x, t; \Theta) dx dt \\ &\quad + \int_0^1 k(x; \Theta) \langle \nabla_{\Theta} k(x; \Theta), \rho \rangle dx. \end{aligned}$$

Taking $\rho = (1, 0)^\top$ gives

$$\nabla_{\theta_1} g_0(\Theta) = - \int_0^T \int_0^1 x v_x(1, t; \Theta) y(x, t; \Theta) dx dt + \frac{1}{3} \theta_1 + \frac{1}{4} \theta_2.$$

Similarly, taking $\rho = (0, 1)^\top$ gives

$$\nabla_{\theta_2} g_0(\Theta) = - \int_0^T \int_0^1 x^2 v_x(1, t; \Theta) y(x, t; \Theta) dx dt + \frac{1}{4} \theta_1 + \frac{1}{5} \theta_2.$$

This completes the proof. □

4 Numerical Solution Procedure

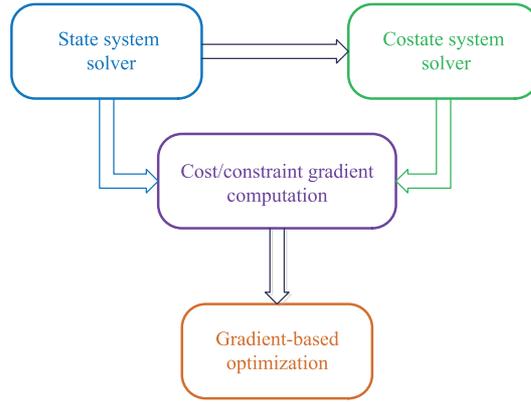
Based on the gradient formulas derived in Section 3, we now propose a gradient-based optimization framework for solving Problem P₂. This framework is illustrated in Figure 2 and described in detail below.

Algorithm 4.1. Gradient-based optimization procedure for solving Problem P₂.

- (a) Choose an initial guess $(\theta_1, \theta_2, \alpha)$.
- (b) Solve the state PDE system (2.3) corresponding to (θ_1, θ_2) .
- (c) Solve the costate PDE system (3.3) corresponding to (θ_1, θ_2) .
- (d) Compute the cost and constraint gradients at $(\theta_1, \theta_2, \alpha)$ using (3.1), (3.2), and (3.4).
- (e) Use the gradient information obtained in Step (d) to perform an optimality test. If $(\theta_1, \theta_2, \alpha)$ is optimal, then stop; otherwise, go to Step (f).
- (f) Use the gradient information obtained in Step (d) to calculate a search direction.
- (g) Perform a line search to determine the optimal step length.
- (h) Compute a new point $(\theta_1, \theta_2, \alpha)$ and return to Step (b).

Note that Steps (e)-(h) of Algorithm 4.1 can be performed automatically by standard nonlinear optimization solvers such as FMINCON in MATLAB.

Recall from Theorem 2.2 that to guarantee closed-loop stability, the optimal value of α must be the *first* positive solution of (2.19). In practice, this can usually be achieved by choosing $\alpha = 0$ as the initial guess. Moreover, after solving Problem P₂, it is easy to check whether the optimal value of α is indeed the smallest positive solution by plotting the left-hand side of (2.19).

Figure 2: Gradient-based optimization framework for solving Problem P₂.

4.1 Simulation of the State System

To solve the state system (2.3) numerically, we will develop a finite-difference method. This method involves discretizing both the spatial and the temporal domains into a finite number of subintervals, i.e.,

$$\begin{aligned} x_0 = 0, x_1 = h, x_2 = 2h, \dots, x_n = nh = 1, \\ t_0 = 0, t_1 = \tau, t_2 = 2\tau, \dots, t_m = m\tau = T, \end{aligned}$$

where n and m are positive integers and $h = 1/n$ and $\tau = T/m$. Using the Taylor expansion, we obtain the following approximations:

$$\frac{\partial y(x_i, t_j)}{\partial t} = \frac{y(x_i, t_j + \tau) - y(x_i, t_j)}{\tau} + \mathcal{O}(\tau), \quad (4.1a)$$

$$\frac{\partial^2 y(x_i, t_j)}{\partial x^2} = \frac{y(x_i + h, t_j) - 2y(x_i, t_j) + y(x_i - h, t_j)}{h^2} + \mathcal{O}(h^2), \quad (4.1b)$$

where $\mathcal{O}(\tau)$ and $\mathcal{O}(h^2)$ denote, respectively, omitted first- and second-order terms such that $\mathcal{O}(\tau) \rightarrow 0$ as $\tau \rightarrow 0$ and $h^{-1}\mathcal{O}(h^2) \rightarrow 0$ as $h \rightarrow 0$. Substituting (4.1) into (2.3a) gives

$$\frac{y_{i,j+1} - y_{i,j}}{\tau} = \frac{y_{i+1,j} - 2y_{i,j} + y_{i-1,j}}{h^2} + cy_{i,j},$$

where $y_{i,j} = y(x_i, t_j)$, $i = 0, 1, \dots, n$, $j = 0, 1, \dots, m$. Simplifying this equation, we obtain

$$y_{i,j+1} = (1 - 2r + c\tau)y_{i,j} + r(y_{i-1,j} + y_{i+1,j}), \quad (4.2)$$

where $1 \leq i \leq n - 1$, $0 \leq j \leq m - 1$ and

$$r = \frac{\tau}{h^2}. \quad (4.3)$$

The explicit numerical scheme (4.2) is convergent when $0 < r \leq 0.5$ (see reference [3] for the relevant convergence analysis). Thus, in this paper, we assume that τ and h are chosen such that $0 < r \leq 0.5$. From (2.3b), we obtain the initial condition

$$y_{i,0} = y(x_i, 0) = y_0(x_i), \quad i = 0, 1, \dots, n. \quad (4.4)$$

Moreover, from (2.3c) and (2.3d), we obtain the boundary conditions

$$y_{0,j} = y(0, t_j) = 0, \quad j = 1, 2, \dots, m, \tag{4.5}$$

and

$$y_{n,j} = y(1, t_j) = \int_0^1 k(\xi; \Theta)y(\xi, t_j)d\xi, \quad j = 1, 2, \dots, m. \tag{4.6}$$

Using the composite trapezoidal rule [3], the integral in (4.6) becomes

$$\begin{aligned} y_{n,j} &= \frac{1}{2}h\{k(x_0; \Theta)y(x_0, t_j) + k(x_n; \Theta)y(x_n, t_j)\} + h \sum_{i=1}^{n-1} k(x_i; \Theta)y(x_i, t_j) \\ &= \frac{1}{2}hk(x_n; \Theta)y_{n,j} + h \sum_{i=1}^{n-1} k(x_i; \Theta)y_{i,j}. \end{aligned}$$

Rearranging this equation yields

$$y_{n,j} = \left[1 - \frac{1}{2}hk(x_n; \Theta)\right]^{-1} \left[h \sum_{i=1}^{n-1} k(x_i; \Theta)y_{i,j}\right]. \tag{4.7}$$

By using the initial condition (4.4) and the boundary conditions (4.5) and (4.7), numerical approximations of $y(x, t)$ at the pre-defined nodes can be calculated forward in time recursively from (4.2).

4.2 Simulation of the Costate System

As with the state system, we will use the finite-difference method to solve the costate system (3.3) numerically. Using the Taylor expansion, we obtain the following approximations:

$$\frac{\partial v(x_i, t_j)}{\partial t} = \frac{v(x_i, t_j) - v(x_i, t_j - \tau)}{\tau} + \mathcal{O}(\tau), \tag{4.8a}$$

$$\frac{\partial^2 v(x_i, t_j)}{\partial x^2} = \frac{v(x_i + h, t_j) - 2v(x_i, t_j) + v(x_i - h, t_j)}{h^2} + \mathcal{O}(h^2), \tag{4.8b}$$

$$\frac{\partial v(1, t_j)}{\partial x} = \frac{v(x_n, t_j) - v(x_n - h, t_j)}{h} + \mathcal{O}(h). \tag{4.8c}$$

Substituting (4.8) into (3.3) gives

$$\frac{v_{i,j} - v_{i,j-1}}{\tau} + \frac{v_{i+1,j} - 2v_{i,j} + v_{i-1,j}}{h^2} + cv_{i,j} + y_{i,j} - k(x_i; \Theta)\frac{v_{n,j} - v_{n-1,j}}{h} = 0,$$

where $v_{i,j} = v(x_i, t_j)$. We rearrange this equation to obtain

$$v_{i,j-1} = (1 - 2r + c\tau)v_{i,j} + r(v_{i+1,j} + v_{i-1,j}) + \tau y_{i,j} - \frac{\tau k(x_i; \Theta)}{h}(v_{n,j} - v_{n-1,j}), \tag{4.9}$$

where $1 \leq i \leq n - 1, 1 \leq j \leq m$ and r is as defined in (4.3). From (3.3c), we obtain the terminal condition

$$v_{i,m} = v(x_i, T) = 0, \quad i = 0, 1, \dots, n. \tag{4.10}$$

Moreover, from (3.3b), we obtain the boundary conditions

$$v_{0,j} = v(0, t_j) = 0, \quad v_{n,j} = v(1, t_j) = 0, \quad j = 0, 1, \dots, m. \tag{4.11}$$

Using the recurrence equation (4.9), together with (4.10) and (4.11), we can compute approximate values of $v(x, t)$ backward in time. The finite-difference schemes for solving the state and costate PDEs are summarized in Table 1.

Table 1: Numerical computation of $y(x, t)$ and $v(x, t)$ **Procedure 1.** Evaluation of $y(x_i, t_j)$.

- 1: Set $1/n \rightarrow h, T/m \rightarrow \tau$.
- 2: Compute $y_{i,0}$ for each $i = 0, 1, \dots, n$ using (4.4).
- 3: Set $1 \rightarrow j$.
- 4: Compute $y_{i,j}$ for each $i = 1, 2, \dots, n-1$ by solving (4.2).
- 5: Compute $y_{0,j}$ using (4.5).
- 6: Compute $y_{n,j}$ using (4.7).
- 7: If $j = m$, then stop. Otherwise, set $j+1 \rightarrow j$ and go to Step 4.

Procedure 2. Evaluation of $v(x_i, t_j)$.

- 1: Compute $v_{i,m}$ for each $i = 0, 1, \dots, n$ using (4.10).
- 2: Set $j = m-1$.
- 3: Compute $v_{i,j}$ for each $i = 1, 2, \dots, n-1$ using (4.9).
- 4: Compute $v_{0,j}$ and $v_{n,j}$ using (4.11).
- 5: If $j = 0$, then stop. Otherwise, set $j-1 \rightarrow j$ and go to Step 3.

4.3 Numerical Integration

Recall the cost functional (2.7):

$$\begin{aligned} g_0(\Theta) &= \frac{1}{2} \int_0^T \int_0^1 y^2(x, t; \Theta) dx dt + \frac{1}{2} \int_0^1 k^2(x; \Theta) dx \\ &= \frac{1}{2} \int_0^T \int_0^1 y^2(x, t; \Theta) dx dt + \frac{1}{6} \theta_1^2 + \frac{1}{10} \theta_2^2 + \frac{1}{4} \theta_1 \theta_2. \end{aligned}$$

Furthermore, recall the cost functional's gradient from (3.4):

$$\begin{aligned} \nabla_{\theta_1} g_0(\Theta) &= - \int_0^T \int_0^1 x v_x(1, t; \Theta) y(x, t; \Theta) dx dt + \frac{1}{3} \theta_1 + \frac{1}{4} \theta_2, \\ \nabla_{\theta_2} g_0(\Theta) &= - \int_0^T \int_0^1 x^2 v_x(1, t; \Theta) y(x, t; \Theta) dx dt + \frac{1}{4} \theta_1 + \frac{1}{5} \theta_2. \end{aligned}$$

Clearly, both the cost functional (2.7) and its gradient (3.4) involve evaluating double integrals of the form

$$\int_0^T \int_0^1 \psi(x, t) dx dt, \tag{4.12}$$

where $\psi(x, t) = y^2(x, t; \Theta)$ for the cost functional and $\psi(x, t) = -\nabla_{\theta_i} k(x; \Theta) v_x(1, t; \Theta) y(x, t; \Theta)$, $i = 1, 2$, for the cost functional's gradient. To evaluate these integrals, we partition the space and temporal domains using the same equally-spaced mesh points x_0, x_1, \dots, x_n and t_0, t_1, \dots, t_m as in Sections 4.1 and 4.2. These subintervals define step sizes $h = 1/n$ and $\tau = T/m$. The integral in (4.12) can be written as the following iterated integral:

$$\int_0^T \int_0^1 \psi(x, t) dx dt = \int_0^T \left(\int_0^1 \psi(x, t) dx \right) dt.$$

Applying the composite Simpson’s rule [3] twice, we obtain the following approximation:

$$\int_0^T \int_0^1 \psi(x, t) dx dt = \frac{1}{3} \tau \left\{ \phi(t_0) + 2 \sum_{l=1}^{(m/2)-1} \phi(t_{2l}) + 4 \sum_{l=1}^{m/2} \phi(t_{2l-1}) + \phi(t_m) \right\}, \quad (4.13)$$

where

$$\phi(t_j) = \frac{1}{3} h \left[\psi(x_0, t_j) + 2 \sum_{k=1}^{(n/2)-1} \psi(x_{2k}, t_j) + 4 \sum_{k=1}^{n/2} \psi(x_{2k-1}, t_j) + \psi(x_n, t_j) \right].$$

More details on numerical integration algorithms are available in [3]. Using (4.13), the cost functional (2.7) and its gradient (3.4) can be evaluated successfully.

5 Numerical Simulations

Our numerical simulations were conducted within the MATLAB programming environment running on a desktop computer with the following configuration: Intel Core i7-2600 3.40GHz CPU, 4.00GB RAM, 64-bit Windows 7 Operating System. For the finite-difference discretization, we used $n = 14$ spatial intervals and $m = 5000$ temporal intervals over a time horizon of $[0, T] = [0, 4]$. Our code implements the gradient-based optimization procedure in Algorithm 4.1 by combining the FMINCON function in MATLAB with the gradient computation method described in Section 3.

Consider the uncontrolled version of (2.3) in which $u(t) = 0$. In this case, the exact solution is

$$y(x, t) = 2 \sum_{n=1}^{\infty} C_n e^{(c-n^2\pi^2)t} \sin(n\pi x) dx, \quad (5.1)$$

where C_n are the Fourier coefficients defined by

$$C_n = \int_0^1 y_0(x) \sin(n\pi x) dx.$$

The eigenvalues of (5.1) are $c - n^2\pi^2$, $n = 1, 2, \dots$. The largest eigenvalue is therefore $c - \pi^2$, which indicates that system (2.3) with $u(t) = 0$ is unstable for $c > \pi^2 \approx 9.8696$. We report the numerical results from our algorithm for three different scenarios.

5.1 Scenario 1

For the first scenario, we choose $c = 10$ and $y_0(x) = \sin(\pi x)$. The corresponding uncontrolled open-loop response (see equation (5.1)) is shown in Figure 3. As we can see from Figure 3, the state of the uncontrolled system grows as time increases. For the feedback kernel optimization, we suppose that the lower and upper bounds for the optimization parameters are $a_i = -10$ and $b_i = 10$, respectively. We also choose $\epsilon = 1$ in (2.22b). Starting from the initial guess $(\theta_1, \theta_2, \alpha) = (-1.0, 2.0, 0)$, our program terminates after 23 iterations and 15.8358 seconds. The optimal cost value is $g_0 = 0.1712$ and the optimal solution of Problem P₂ is $(\theta_1^*, \theta_2^*, \alpha^*) = (-1.0775, 0.5966, 3.3486)$.

The spatial-temporal response of the controlled plant corresponding to (θ_1^*, θ_2^*) is shown in Figure 4(a). The figure clearly shows that the controlled system (2.3) with optimized parameters (θ_1^*, θ_2^*) is stable. The corresponding boundary control and kernel function are shown in Figures 4(b) and 4(c), respectively.

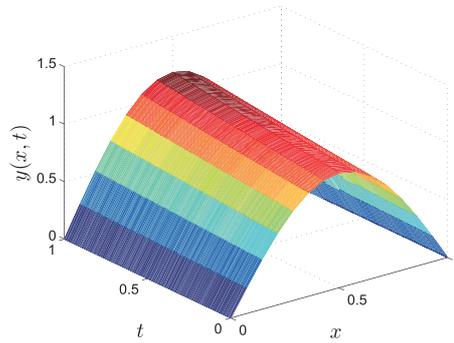


Figure 3: Uncontrolled open-loop response for Scenario 1.

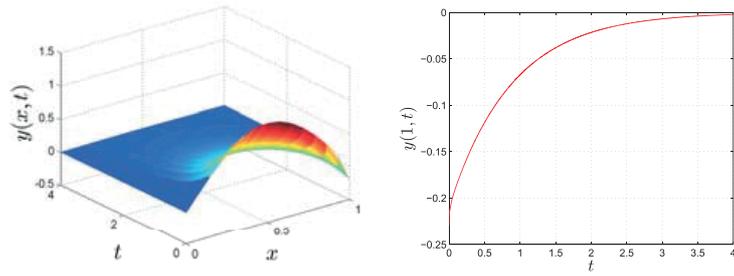
Table 2: Solutions of (2.19) and corresponding optimal span coefficients Y_n in (2.23) for Scenario 1.

n	α_n^*	α_n^*/π	Y_n
1	3.3486	1.0658	1.0364
2	6.3838	2.0320	-0.0915
3	9.4952	3.0224	0.0505
4	12.6173	4.0162	-0.0360
5	15.7493	5.0131	0.0268
6	18.8835	6.0108	-0.0206
7	22.0205	7.0093	0.0161
8	25.1582	8.0081	-0.0126
9	28.2971	9.0072	0.0098
10	31.4363	10.0064	-0.0074

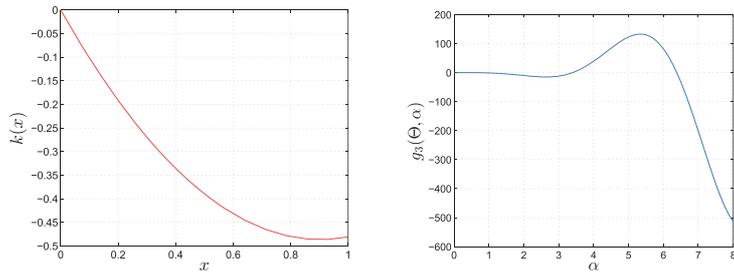
Recall from Theorem 2.2 that closed-loop stability is guaranteed if $\alpha^* = 3.3486$ is the first positive solution of equation (2.19) and the initial function $y_0(x)$ is contained within the linear span of $\{\sin(\alpha_n^* x)\}$, where each α_n^* is a solution of equation (2.19) corresponding to (θ_1^*, θ_2^*) . It is clear from Figure 4(d) that α^* is indeed the first positive solution of equation (2.19). To verify the linear span condition, we use FMINCON in MATLAB to minimize (2.23) for $N = 10$. The first 10 positive solutions of (2.19) corresponding to the optimal parameters $\theta_1^* = -1.0775$ and $\theta_2^* = 0.5966$ are given in Table 2. The optimal span coefficients that minimize (2.23) are also given. The optimal value of J in (2.23) is 2.184832×10^{-5} , which indicates that the span condition holds. Note also from Table 2 that α_n^*/π converges to an integer as $n \rightarrow \infty$ (recall the discussion of the end of Section 2.2).

5.2 Scenario 2

For the second scenario, we choose $c = 11$ and $y_0(x) = (1+x)\sin(\pi x)$. The corresponding uncontrolled open-loop trajectory is shown in Figure 5. Starting from the initial guess $(\theta_1, \theta_2, \alpha) = (-1.0, 1.5, 0)$, our program converges after 26 iterations and 11.5767 seconds with an optimal cost value of $g_0 = 0.5515$. The corresponding optimal parameter values are $(\theta_1^*, \theta_2^*, \alpha^*) = (-2.9141, 1.7791, 3.6056)$. We show the spatial-temporal response for the controlled system with optimized feedback parameters (θ_1^*, θ_2^*) in Figure 6(a). Again, as with Scenario 1, the controlled plant corresponding to the optimal solution of Problem P₂ is



(a) Closed-loop response $y(x, t)$. (b) Optimal boundary control $y(1, t)$.



(c) Optimal kernel $k(x)$. (d) Left-hand side of (2.19).

Figure 4: Simulation results for Scenario 1 (optimized parameters $\theta_1^* = -1.0775$, $\theta_2^* = 0.5966$, $\alpha^* = 3.3486$).

stable. The optimal boundary control and optimal kernel function are shown in Figures 6(b) and 6(c), respectively. Figure 6(d) shows the left-hand side of (2.19). Note that $\alpha^* = 3.6056$ is the first positive root, as required by Theorem 2.2. Using MATLAB to minimize (2.23) for $N = 14$, we obtain an optimal cost of 1.249410×10^{-12} , which indicates that the span condition in Theorem 2.2 holds. The values of α_n^* and Y_n in (2.23) are given in Table 3.

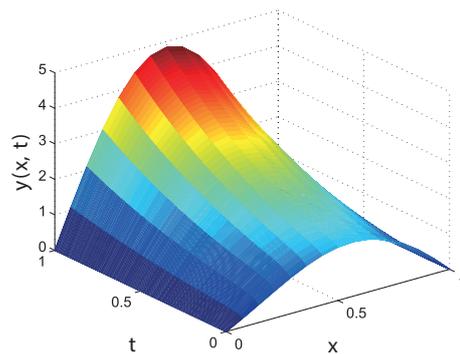


Figure 5: Uncontrolled open-loop response for Scenario 2.

Table 3: Solutions of (2.19) and optimal span coefficients Y_n in (2.23) for Scenario 2.

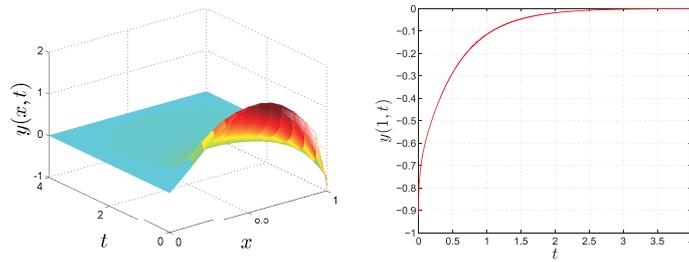
n	α_n^*	α_n^*/π	Y_n
1	3.6056	1.1476	-0.4185
2	6.4595	2.0562	1.4867
3	9.5520	3.0404	1.4965
4	12.6561	4.0285	-0.7676
5	15.7817	5.0234	0.4462
6	18.9096	6.0191	-0.3391
7	22.0433	7.0166	0.2493
8	25.1778	8.0143	-0.1992
9	28.3147	9.0128	0.1520
10	31.4520	10.0114	-0.1189
11	34.5905	11.0104	0.0871
12	37.7291	12.0095	-0.0622
13	40.8685	13.0088	0.0429
14	44.0081	14.0086	1.0061

5.3 Scenario 3

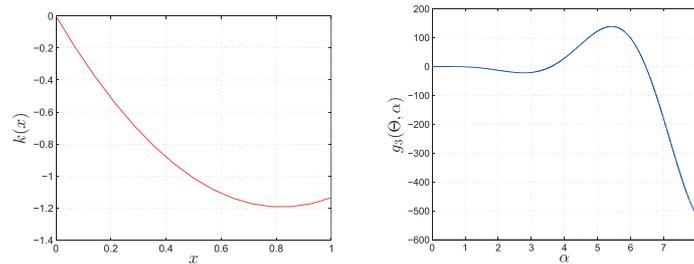
For the final scenario, we choose $c = 14$ and $y_0(x) = (2 + x)\sin(2.5\pi x)$. The corresponding uncontrolled open-loop trajectory is shown in Figure 7. Starting from the initial guess $(\theta_1, \theta_2, \alpha) = (-2.0, 1.5, 0)$, our program terminates after 22 iterations and 10.0226 seconds with an optimal cost value of $g_0 = 3.1006$. The corresponding optimal solution is $(\theta_1^*, \theta_2^*, \alpha^*) = (-9.1266, 6.4093, 4.1231)$. The spatial-temporal response of the controlled plant corresponding to (θ_1^*, θ_2^*) is shown in Figure 8(a), which clearly shows that the controlled system (2.3) with optimized parameters (θ_1^*, θ_2^*) is stable. The optimal boundary control and optimal kernel function are shown in Figures 8(b) and 8(c), respectively. Minimizing (2.23) for $N = 14$ yields an optimal cost of 8.045397×10^{-15} . We report the corresponding values of α_n^* and Y_n in Table 4. Finally, Figure 8(d) shows the left-hand side of equation (2.19) corresponding to the optimized parameters.

6 Conclusion

In this paper, we have introduced a new gradient-based optimization approach for boundary stabilization of parabolic PDE systems. As with the well-known LQ control and backstepping synthesis approaches, our new approach involves expressing the boundary controller as an integral state feedback in which a kernel function needs to be designed judiciously. However, unlike the LQ control and backstepping approaches, we do not determine the feedback kernel by solving Riccati-type or Klein-Gorden-type PDEs; instead, we approximate the feedback kernel by a quadratic function and then optimize the quadratic's coefficients using dynamic optimization techniques. This approach requires solving a so-called "costate PDE", which is much easier to solve numerically than the Riccati and Klein-Gorden PDEs. Indeed, as shown in Section 4, the costate PDE can be solved easily using the finite-difference method. Based on the work in this paper, we have identified several unresolved research questions described as follows: (i) Is it possible to prove, or at least weaken, the linear span condition in Theorem 2.2? (ii) Can the proposed kernel optimization approach be applied to other classes of PDE plant models? (iii) Is it possible to develop methods for minimizing cost functional (2.7) over an infinite time horizon? These issues will be explored in future work.



(a) Closed-loop response $y(x, t)$. (b) Optimal boundary control $y(1, t)$.



(c) Optimal kernel $k(x)$. (d) Left-hand side of (2.19).

Figure 6: Simulation results for Scenario 2 (optimized parameters $\theta_1^* = -2.9141$, $\theta_2^* = 1.7791$, $\alpha^* = 3.6056$).

Acknowledgements

This paper is dedicated to Professor Kok Lay Teo on the occasion of his 70th birthday. Professor Teo has been valued colleague and mentor to each of the authors of this paper. The gradient computation procedure in Section 3 was inspired by Professor Teo’s seminal work in [19]; see also the recent survey papers [11, 12].

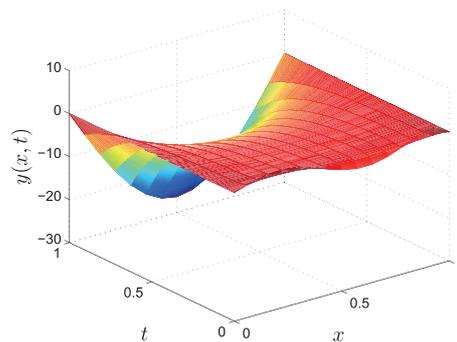
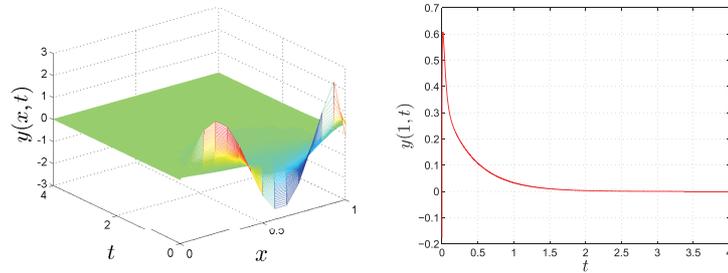


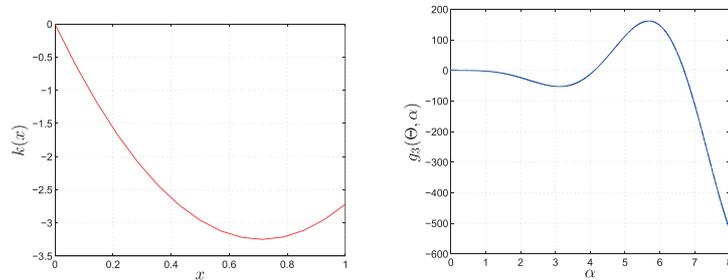
Figure 7: Uncontrolled open-loop response for Scenario 3.

Table 4: Solutions of (2.19) and corresponding optimal span coefficients Y_n in (2.23) for Scenario 3.

n	α_n^*	α_n^*/π	Y_n
1	4.1231	1.3124	-0.4383
2	6.6959	2.1314	1.7274
3	9.7345	3.0986	1.2549
4	12.7804	4.0683	-0.7124
5	15.8861	5.0564	0.4225
6	18.9930	6.0456	-0.3256
7	22.1166	7.0399	0.2415
8	25.2406	8.0343	-0.1951
9	28.3713	9.0308	0.1510
10	31.5022	10.0274	-0.1208
11	34.6366	11.0251	0.0922
12	37.7711	12.0229	-0.0723
13	40.9075	13.0212	0.0662
14	44.0440	14.0196	1.0151



(a) Closed-loop response $y(x, t)$. (b) Optimal boundary control $y(1, t)$.



(c) Optimal kernel $k(x)$. (d) Left-hand side of (2.19).

Figure 8: Simulation results for Scenario 3 (optimized parameters $\theta_1^* = -9.1266$, $\theta_2^* = 6.4093$, $\alpha^* = 4.1231$).

A Proof of Lemma 2.1

We prove the lemma in three steps.

A.1 Preliminaries

Let

$$Q(\alpha) = \sqrt{(\theta_1\alpha^2 + \theta_2\alpha^2 - 2\theta_2)^2 + (\alpha^3 - \theta_1\alpha - 2\theta_2\alpha)^2}$$

$$= \sqrt{\alpha^6 + (\theta_1^2 + 2\theta_1\theta_2 + \theta_2^2 - 2\theta_1 - 4\theta_2)\alpha^4 + \theta_1^2\alpha^2 + 4\theta_2^2}.$$

Furthermore, let $\varphi(\alpha) \in (-\pi, \pi]$ be the unique angle satisfying

$$\cos(\varphi(\alpha)) = \frac{\alpha^3 - \theta_1\alpha - 2\theta_2\alpha}{Q(\alpha)},$$

and

$$\sin(\varphi(\alpha)) = \frac{\theta_1\alpha^2 + \theta_2\alpha^2 - 2\theta_2}{Q(\alpha)}.$$

Using the definitions of $Q(\alpha)$ and $\varphi(\alpha)$, equation (2.19) can be rewritten as follows:

$$Q(\alpha) \sin(\varphi(\alpha)) \cos(\alpha) + Q(\alpha) \cos(\varphi(\alpha)) \sin(\alpha) = -2\theta_2.$$

Thus, using the angle sum trigonometric identity, we obtain

$$Q(\alpha) \sin(\alpha + \varphi(\alpha)) = -2\theta_2. \tag{A.1}$$

Now, under condition (2.20), $Q(\alpha) \rightarrow \infty$ as $\alpha \rightarrow \infty$. Furthermore,

$$\lim_{\alpha \rightarrow +\infty} \cos(\varphi(\alpha)) = 1, \quad \lim_{\alpha \rightarrow +\infty} \sin(\varphi(\alpha)) = 0.$$

Hence, $\varphi(\alpha) \rightarrow 0$ as $\alpha \rightarrow \infty$.

A.2 Angle $\varphi(\alpha)$ is Continuous at all Sufficiently Large α

Since $\varphi(\alpha) \rightarrow 0$ as $\alpha \rightarrow \infty$, there exists a constant $\bar{\alpha}$ such that $-\frac{1}{4}\pi < \varphi(\alpha) < \frac{1}{4}\pi$ for all $\alpha > \bar{\alpha}$. Consider an arbitrary point $\alpha' > \bar{\alpha}$. We will show that $\varphi(\cdot)$ is continuous at α' .

Let $\delta > 0$. In view of the definition of $\sin(\varphi(\alpha))$, there exists an $\varepsilon > 0$ such that

$$|\alpha - \alpha'| < \varepsilon \implies -\frac{1}{\sqrt{2}}\delta < \sin(\varphi(\alpha)) - \sin(\varphi(\alpha')) < \frac{1}{\sqrt{2}}\delta. \tag{A.2}$$

Now, using Taylor's Theorem,

$$\sin(\varphi(\alpha)) - \sin(\varphi(\alpha')) = \cos(\zeta)(\varphi(\alpha) - \varphi(\alpha')), \tag{A.3}$$

where ζ belongs to the interval bounded by $\varphi(\alpha)$ and $\varphi(\alpha')$. Suppose α satisfies $|\alpha - \alpha'| < \min(\varepsilon, \alpha' - \bar{\alpha})$. Then

$$-\frac{1}{4}\pi < \varphi(\alpha) < \frac{1}{4}\pi, \quad -\frac{1}{4}\pi < \varphi(\alpha') < \frac{1}{4}\pi.$$

Hence,

$$-\frac{1}{4}\pi < \zeta < \frac{1}{4}\pi$$

and

$$\cos \zeta > \frac{1}{\sqrt{2}}. \tag{A.4}$$

Combining (A.2)-(A.4) yields

$$\frac{1}{\sqrt{2}}\delta > |\sin(\varphi(\alpha)) - \sin(\varphi(\alpha'))| = |\cos(\zeta)| \cdot |\varphi(\alpha) - \varphi(\alpha')| \geq \frac{1}{\sqrt{2}}|\varphi(\alpha) - \varphi(\alpha')|.$$

Hence, we have established the following implication:

$$|\alpha - \alpha'| < \min(\varepsilon, \alpha' - \bar{\alpha}) \implies |\varphi(\alpha) - \varphi(\alpha')| < \delta.$$

This shows that $\varphi(\cdot)$ is continuous at α' , as required.

A.3 Roots of Equation (A.1)

Let $\epsilon \in (0, \frac{1}{2}\pi)$ and define

$$a_k = k\pi - \epsilon, \quad b_k = k\pi + \epsilon.$$

Clearly, for each integer $k \geq 0$, $a_k < b_k < a_{k+1}$ and

$$\sin(a_k) = \begin{cases} \sin(\epsilon), & \text{if } k \text{ is odd,} \\ -\sin(\epsilon), & \text{if } k \text{ is even,} \end{cases}$$

$$\sin(b_k) = \begin{cases} -\sin(\epsilon), & \text{if } k \text{ is odd,} \\ \sin(\epsilon), & \text{if } k \text{ is even.} \end{cases}$$

Using Taylor's Theorem, we have

$$\sin(\alpha + \varphi(\alpha)) = \sin(\alpha) + \cos(\zeta)\varphi(\alpha),$$

where $\zeta = \zeta(\alpha)$ belongs to the interval bounded by α and $\alpha + \varphi(\alpha)$. Thus,

$$|\sin(\alpha + \varphi(\alpha)) - \sin(\alpha)| = |\cos(\zeta)\varphi(\alpha)| \leq |\varphi(\alpha)|. \quad (\text{A.5})$$

Since $\varphi(\alpha) \rightarrow 0$ as $\alpha \rightarrow \infty$, there exists an integer $k_1 \geq 1$ such that for all $k \geq k_1$,

$$|\varphi(a_k)| < \frac{1}{2}\sin(\epsilon), \quad |\varphi(b_k)| < \frac{1}{2}\sin(\epsilon).$$

Hence, substituting $\alpha = a_k$ and $\alpha = b_k$ into (A.5) gives, for $k \geq k_1$,

$$\sin(a_k + \varphi(a_k)) \begin{cases} > \frac{1}{2}\sin(\epsilon), & \text{if } k \text{ is odd,} \\ < -\frac{1}{2}\sin(\epsilon), & \text{if } k \text{ is even,} \end{cases}$$

$$\sin(b_k + \varphi(b_k)) \begin{cases} < -\frac{1}{2}\sin(\epsilon), & \text{if } k \text{ is odd,} \\ > \frac{1}{2}\sin(\epsilon), & \text{if } k \text{ is even.} \end{cases}$$

Since $Q(\alpha) \rightarrow \infty$ as $\alpha \rightarrow \infty$, there exists an integer $k_2 \geq 1$ such that for all $k \geq k_2$,

$$-\frac{1}{2}Q(a_k)\sin(\epsilon) \leq -2\theta_2 \leq \frac{1}{2}Q(a_k)\sin(\epsilon),$$

$$-\frac{1}{2}Q(b_k)\sin(\epsilon) \leq -2\theta_2 \leq \frac{1}{2}Q(b_k)\sin(\epsilon).$$

Thus, for all $k \geq \max\{k_1, k_2\}$,

$$Q(a_k) \sin(a_k + \varphi(a_k)) \begin{cases} > \frac{1}{2}Q(a_k) \sin(\epsilon) \geq -2\theta_2, & \text{if } k \text{ is odd,} \\ < -\frac{1}{2}Q(a_k) \sin(\epsilon) \leq -2\theta_2, & \text{if } k \text{ is even,} \end{cases}$$

$$Q(b_k) \sin(b_k + \varphi(b_k)) \begin{cases} < -\frac{1}{2}Q(b_k) \sin(\epsilon) \leq -2\theta_2, & \text{if } k \text{ is odd,} \\ > \frac{1}{2}Q(b_k) \sin(\epsilon) \geq -2\theta_2, & \text{if } k \text{ is even.} \end{cases}$$

Since φ is continuous when α is large, this implies that, for all sufficiently large k , there exists a solution of (A.1) within the interval $[a_k, b_k]$. The result follows immediately.

References

- [1] A.C. Antoulas, *Approximation of Large-scale Dynamical Systems*, SIAM, Philadelphia, 2005.
- [2] A. Bensoussan, G. Da Prato, M.C. Delfour and S.K. Mitter, *Representation and Control of Infinite Dimensional Systems*, Birkhauser, Boston, 2007.
- [3] R.L. Burden and J.D. Faires, *Numerical Analysis*, Cengage Learning, Boston, 1993.
- [4] R.F. Curtain and H. Zwart, *An Introduction to Infinite-dimensional Linear Systems Theory*, Springer, New York, 1995.
- [5] J.H. Ferziger and M. Perić, *Computational Methods for Fluid Dynamics*, Springer, Berlin, 1996.
- [6] P. Holmes, J.L. Lumley and G. Berkooz, *Turbulence, Coherent Structures, Dynamical Systems and Symmetry*, Cambridge University Press, London, 1998.
- [7] N.J. Killingsworth and M. Krstic, PID tuning using extremum seeking: Online, model-free performance optimization, *IEEE Control Syst. Mag.* 26 (2006) 70–79.
- [8] M. Krstic, B. Guo, A. Balogh and A. Smyshlyaev, Control of a tip-force destabilized shear beam by observer-based boundary feedback, *SIAM J. Control Optim.* 47 (2008) 553–574.
- [9] M. Krstic and A. Smyshlyaev, *Boundary Control of PDEs: A Course on Backstepping Designs*, SIAM, Philadelphia, 2008.
- [10] B. Li, K. L. Teo, C. Lim and G. Duan, An optimal PID controller design for nonlinear constrained optimal control problems, *Discrete Contin. Dyn. Syst. Ser. B* 16 (2011) 70–79.
- [11] Q. Lin, R. Loxton and K.L. Teo, Optimal control of nonlinear switched systems: Computational methods and applications, *J. Oper. Res. Soc. China* 1 (2013) 275–311.
- [12] Q. Lin, R. Loxton and K.L. Teo, The control parameterization method for nonlinear optimal control: A survey, *J. Ind. Manag. Optim.* 10 (2014) 275–309.
- [13] W. Liu, Boundary feedback stabilization of an unstable heat equation, *SIAM J. Control Optim.* 42 (2003) 1033–1043.

- [14] W. Liu, *Elementary Feedback Stabilization of the Linear Reaction-convection-diffusion Equation and the Wave Equation*, Springer, Berlin, 2010.
- [15] B. Moore, Principal component analysis in linear systems: Controllability, observability, and model reduction, *IEEE Trans. Automat. Control* 26 (1981) 17–32.
- [16] S.J. Moura and H.K. Fathy, Optimal boundary control of reaction-diffusion partial differential equations via weak variations, *ASME J. Dyn. Syst. Meas. Control* 135 (2013) 034501(1-8).
- [17] J. Strikwerda, *Finite Difference Schemes and Partial Differential Equations*, SIAM, Philadelphia, 2007.
- [18] G.A. Susto and M. Krstic, Control of PDE-ODE cascades with Neumann interconnections, *J. Franklin Inst.* 347 (2010) 284–314.
- [19] K.L. Teo, C.J. Goh and K.H. Wong, *A Unified Computational Approach to Optimal Control Problems*, Longman Scientific and Technical, Essex, 1991.
- [20] R. Triggiani, Well-posedness and regularity of boundary feedback parabolic systems, *J. Differential Equations* 36 (1980) 347–362.
- [21] R. Vazquez and M. Krstic, A closed-form feedback controller for stabilization of the linearized 2-D Navier-Stokes poiseuille system, *IEEE Trans. Automat. Control* 52 (2007) 2298–2312.
- [22] R. Vazquez and M. Krstic, *Control of Turbulent and Magnetohydrodynamic Channel Flows: Boundary Stabilization and State Estimation*, Springer, Boston, 2008.
- [23] J. Xu, D. Huang and S. Pindi, Optimal tuning of PID parameters using iterative learning approach, *SICE J. Control Meas. Syst. Integr.* 1 (2008) 143–154.
- [24] C. Xu, Y. Ou and E. Schuster, Sequential linear quadratic control of bilinear parabolic PDEs based on POD model reduction, *Automatica J. IFAC* 47 (2011) 418–426.
- [25] C. Xu, E. Schuster, R. Vazquez and M. Krstic, Stabilization of linearized 2D magnetohydrodynamic channel flow by backstepping boundary control, *Systems Control Lett.* 57 (2008) 805–812.

Manuscript received 23 August 2014
revised 21 December 2014
accepted for publication 27 February 2015

ZHIGANG REN
Institute of Cyber-Systems and Control
Zhejiang University, Hangzhou, China
E-mail address: zhigren@hotmail.com

CHAO XU
Institute of Cyber-Systems and Control
Zhejiang University, Hangzhou, China
E-mail address: cxu@zju.edu.cn

QUN LIN

Department of Mathematics and Statistics
Curtin University, Perth, Australia
E-mail address: q.lin@curtin.edu.au

RYAN LOXTON

Department of Mathematics and Statistics
Curtin University, Perth, Australia
E-mail address: R.Loxton@curtin.edu.au