A Method for Approximating and Controlling the Distributed Parameter Model of a MIMO Chemical Diffusion System

by

Alisha R. Schor

S.B., Massachusetts Institute of Technology (2007)

Submitted to the Department of Mechanical Engineering in partial fulfillment of the requirements for the degree of Master of Science in Mechanical Engineering at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2010

© Massachusetts Institute of Technology 2010. All rights reserved.

Author

Department of Mechanical Engineering

May 7, 2010

Certified by

H. Harry Asada
Ford Professor of Mechanical Engineering
Thesis Supervisor

Accepted by

David Hardt
Chairman, Department Committee on Graduate Theses
Abstract

Chemical distribution is an important factor in many biological systems. Chemotaxis, the directed movement of an organism in response to chemical in its environment, is known to be a prominent driver in the directed growth of new blood vessels from pre-existing ones, a phenomenon called angiogenesis. In order to properly study the effects of various chemical inputs to an angiogenic assay, it is crucial to have strict control over the delivery of these chemicals, which are carried to the sprouting site via diffusion. More specifically, we use, as a model system, a microfluidic, in vitro assay in which a cell scaffold is bounded between two microchannels. The scaffold is taken to be a porous region through which diffusion occurs, in one dimension, from one channel to the other. In this system, we can specify the chemical concentration and gradient within the region by changing the concentrations in the channels that bound it. In control terms, this is a multi-input, multi-output (MIMO) system, with two inputs and two outputs. However, the dynamics of diffusion are governed by a partial differential equation, meaning that the plant in question is a distributed parameter system, not immediately amenable to controller design. Thus, in this thesis, we present a method for transforming the diffusion equation (Fick’s Law) into a finitely-approximated, MIMO, state-space system, by first deriving a matrix of infinite transfer functions. With this state-space system, it is shown that classical control techniques can be applied to manipulate the dynamics as well as track a reference step input with zero steady-state error.

Thesis Supervisor: H. Harry Asada
Title: Ford Professor of Mechanical Engineering
Today I choose to write my thanks in verse,
and mention those who made this work come true.
Without their guidance; t'would have been much worse.
So Mom and Dad, this firstly is for you.

Adept advising does a thesis take,
and labmates have their part in helping out.
I thank those friends who ever brought me cake,
tossed disc with me or assuaged my doubts.

To the department admins I owe much,
a more supportive staff cannot be found.
Continuing, I'd say that I am touched,
by friends, and fam; so good to have you 'round.

So in the end, I just want to say thanks,
to all who've helped me join the master's ranks.
Contents

1 Introduction ................................................. 11
   1.1 Biological Motivation ............................. 11
       1.1.1 Chemical gradients in biology ........... 12
       1.1.2 Sprouting angiogenesis .................... 12
   1.2 The need for a control problem .................. 13

2 System and hardware .......................................... 17
   2.1 Static microfluidic system ....................... 17
   2.2 Sensors ............................................. 18
       2.2.1 LabVIEW image analysis ................. 20
   2.3 Actuators ........................................... 20
       2.3.1 Mixer ........................................ 22
       2.3.2 Pump ......................................... 24
   2.4 System limitations ............................... 24

3 Plant Description ............................................ 27
   3.1 Governing physics: chemical diffusion .......... 27
       3.1.1 Solutions to the PDE .................... 28
       3.1.2 Role of reaction terms .................. 29
   3.2 The significance of a PDE in control systems .. 30

4 Conversion of a PDE to a control system model ........ 33
   4.1 Prior work and existing methods ............... 33
A.2 Transfer function $G_{21}$ ........................................ 72
A.3 Transfer function $G_{22}$ ........................................ 73

B Partial Fraction Expansions ........................................ 75
B.1 Partial Fraction Expansion for $G_{11}$ ......................... 75
B.2 Partial Fraction Expansion for $G_{21}$ ......................... 76
B.3 Partial Fraction Expansion for $G_{22}$ ......................... 77
Chapter 1

Introduction

1.1 Biological Motivation

In this thesis we present a method for implementing closed loop control in order to specify both concentration and gradient of a chemical within a porous region of a microfluidic device (MFD). While this control problem may arise in number of situations, the system of interest is a polydimethylsiloxane (PDMS) device used to study sprouting angiogenesis [6], a process to be described later in this chapter. In this device, liquid (unpolymerized) collagen is injected into a three-dimensional region and allowed to polymerize into a porous gel. This gel becomes a scaffold for cell growth and with this setup, network-level, three-dimensional response of a group of cells to a variety of chemical and mechanical inputs can be studied. The porosity of the gel allows for the transport of fluids and chemicals through the scaffold.

This porous region is a rectangular prism with a small vertical dimension relative to the height and width of the region. Owing to this and the symmetry of the setup, an essentially one-dimensional gradient can be established via diffusion by introducing differing concentrations of chemical in the microfluidic channels that bound opposing sides the gel region. The remaining two sides are enclosed by the PDMS, considered a solid boundary. In this system, chemical inputs (absolute concentration as well as spatial gradient) are delivered to the scaffold/gel region in open-loop [6], and in fact, many other microfluidic assays rely on similar methods to generate gradients
While these methods have shown to be reliable, they provide little provision for accommodating disturbances or parameter uncertainties. Thus, in this thesis, we will explore the use of feedback control to generate arbitrary and time-varying concentrations and gradients at a specified point in the gel region described above, for the purposes of studying chemical inputs to the angiogenesis process.

1.1.1 Chemical gradients in biology

Chemical gradients are known to play an important role in directing many biological processes, including bacterial motility [2], neutrophil (white blood cell) migration toward a wound [22], directed neuron growth [31] and angiogenesis—the growth of new blood vessels from existing ones [13]. This behavior, in which organisms move either up or down a chemical gradient, is called chemotaxis. From a biologist’s perspective, chemotaxis is a well-studied and reasonably understood phenomenon; it is known, for example, that bacteria will move toward a chemoattractant, and that they sense such gradients by comparing receptor signals along the body length [1]. In other contexts, however, this effect is still not strictly quantified, largely in part due to the fact that the chemical gradients themselves are not often measured precisely. Having strict control over gradients in vitro would greatly improve the study of these behaviors.

1.1.2 Sprouting angiogenesis

As mentioned previously, of particular interest is the study of sprouting angiogenesis. Briefly, angiogenesis is the process through which endothelial cells (ECs), which comprise the walls of blood vessels, break out from the blood vessel wall and begin to form new capillaries, blood vessels, and ultimately blood vessel networks, in response to some stimulus [24]. “Sprouting angiogenesis” refers to only the very early stages, from the breaking out of ECs to the formation of a contiguous tube called a lumen. Angiogenesis plays a crucial role in the vascularization of cancer tumors [24, 33] as well as in certain diseases such as arthritis and some cardiovascular diseases related to atherosclerotic plaques [12], and as a result, is a very widely-studied and scientifically
significant behavior.

It is well-documented that the morphology of angiogenic sprouts is heavily influenced by the presence of angiogenic growth factors, such as vascular endothelial growth factor (VEGF) and fibroblast growth factor (FGF) [14, 24, 33]. Growth inhibitors—angiostatin, angiopoietin I & II and platelet factor-4 (PF4) are also known to play a role [5, 24]. For each of these chemicals, both the absolute concentration and gradient are significant [13].

1.2 The need for a control problem

It is clear that both concentration and gradient of chemicals are of great importance in biology, and as inputs to the process of sprouting angiogenesis specifically. What lacks in current studies, however, is a method for providing these inputs in a specified, time-varying and robust manner. As the study of angiogenesis shifts from a pure biology to a biological engineering perspective, data will become more quantifiable and inputs will accordingly need to be more quantitative. Previous chemotaxis assays such as the Boyden Chamber or Dunn Chamber cannot meet these needs.

Many of the more recent developments in microfluidics have shown that they can provide stable and quantifiable inputs over long periods of time. The Whitesides group used a cascade of fluidic resistances to produce a spatial gradient across a 1mm channel [19]. Using the device shown in Figure 1.1, gradients of various profiles and amplitudes and even dynamic gradients were created.

This device, however, does not suit the needs described above in two regards. Firstly, the device creates the gradients within an open channel, not across a scaffold-filled region. Gradient generation within a gel is complicated by limited actuation sites, since only the region boundaries are accessible. Secondly, the gradients are created in open-loop, meaning that parameter uncertainties as well as disturbances—from the organism under study, from defects in the microfluidic system, etc.—are not compensated for.

Finally, perhaps the most promising use of feedback control is to achieve com-
Figure 1.1: Example configuration of a microfluidic device capable of generating gradients. Reproduced from [19].
Figure 1.2: Possible space of steady-state concentration profiles (shaded gray) for a sample desired position and concentration (black dot). Curved black lines are two concentration profiles that can be achieved transiently, exemplifying the larger space of concentration and gradient profiles.


definitions of concentrations and gradients that are not reachable in the open loop steady-state. Figure 1.2 compares the concentration profiles that can be achieved in the steady state with a pair of sample transient profiles, for a given position and concentration. In open loop, only the gradients associated with steady-state profiles (shaded gray area) can be specified, but with feedback control, we can exploit the much larger space of gradients that arise from the transient cases.

In what follows, we will attempt to overcome the limits of the open loop system by using feedback control to specify the concentration and gradient of a chemical at a particular point within the scaffold, using VEGF as an example chemical of interest. Actuation occurs via the two channels that bound the growth region. As a control problem, this is a two-input, two-output system. It will be shown that this problem is inherently stable in the most simple case; however, closed-loop control can improve the dynamics of the system, reject disturbances and overcome parameter uncertainty, as well as possibly achieve combinations of concentration and gradient that are not achievable in the open-loop configuration.
Chapter 2

System and hardware

2.1 Static microfluidic system

In the previous chapter we noted that there are several existing methods for studying the effects of chemicals and gradients on biological systems. In particular, a microfluidic assay for studying sprouting angiogenesis was developed in the Kamm group at MIT, shown in Figure 2.1. This microfluidic device (MFD) has been designed to study angiogenesis by observing the three-dimensional growth into a porous scaffold that interacts with both the cells and chemical inputs. The devices are fabricated by pouring polydimethylsioxane (PDMS) over a mold created by wet etching a silicon wafer, a process called soft lithography [6, 32]. The feature of interest in this device is the small region at the center, which is filled with a collagen gel that is then allowed to polymerize, leaving a porous scaffold in which cells can grow. This region takes the shape of a rectangular prism that is bounded on top, bottom, and two sides by PDMS, a liquid- and solid-impermeable boundary. The remaining two sides are open to the channels such that the scaffold can interact with cells and media contained in the channels.

The dimensions each region in this representative system are 600µm×1300µm×120µm (W×L×H), with the width defined as the dimension along the channel and length as the dimension in the direction that diffusion occurs, i.e. the distance between channels. Assuming symmetry in the width direction (again, parallel to the channel)
and that the height is small compared to the length, we can view this as a one-dimensional problem. The result is that of the system depicted in Fig. 2.1, the only relevant characteristic is the length of the gel region, $L=1300\mu m$. Our control system "plant", then, is a single slice of scaffold of length $L$, in which the chemical diffuses with diffusion coefficient $D$. The concentration at the endpoints of this slice are manipulated by changing the concentration of chemical in the channels that bound the region, which interface directly with the scaffold.

### 2.2 Sensors

Optical sensing—i.e. a digital camera attached to a microscope—is ubiquitous in microfluidics, due to the fact that many systems are used for cell growth and already require microscope data to understand the biological system. Furthermore, embedding sensors into microfluidics can be difficult and limited, while the wealth of information present in a single image can capture the same data as multiple sensors. For this reason, we consider a CCD camera as our sensor for this control problem,
Figure 2.2: Sample fluorescence image of the scaffold region, with the extracted profile along the line shown. Completely black regions are PDMS; a PDMS post sits within the scaffold region for structural support. Diffusion occurs from the channel at the top into the gel below.

with the images processed in real-time using LabVIEW. Concentration of chemical throughout the scaffold region can be detected using a fluorescence microscope, provided that the chemical is fluorescently tagged. Fluorescent isoforms of VEGF do exist [3], and while it is not entirely known what effect the tagging has on cell behavior, for the purposes of designing a control system for the chemical behavior, we will assume here that we have access to the chemical concentration in this way.

Using this configuration, the resulting measurement is an image of the scaffold region in which the fluorescence intensity indicates concentration. Calibration and background subtraction is required for each experiment to relate a known concentration with a measured intensity level. An example of such an image is shown in Figure 2.2. In this test case, FITC Dextran (40 kDa) was used as a VEGF pseudo-molecule and diffusion was from the channel (top) through the gel region into the opposing channel (at bottom, not shown), in the same device as shown in Fig. 2.1. In order to extract the concentration profile from this image, the grayscale intensity values for each pixel are read and mapped to a concentration level based on the calibration for
that experiment. Pixel numbers are scaled to positions in microns depending on the magnification. (In the image in Fig. 2.2, for example, 0.75 pixels = 1 μm.) With these two pieces of information, a concentration vs. position plot can easily be generated for any position along the width of the scaffold region. Either a sample profile or an average of many can be used as the profile used to compare to the reference.

### 2.2.1 LabVIEW image analysis

Feedback control requires that either the states or outputs be measured in real-time in order to be compared to some reference. In our system, the reference signal is a desired concentration and gradient at a particular point within the diffusion region. This will be discussed in more detail in Sec. 5.1.

While many imaging programs can trivially extract the intensity profile from an image, the key step in implementation is real time processing and interfacing with a controller. LabVIEW 9.0 (National Instruments, Texas, USA) both has an image processing toolkit and is commonly used as an interface for controllers. The LabVIEW toolkit can process either stills or live video feeds from any USB-enabled camera and output an intensity profile. With this information, it is straightforward to calculate both concentration and (spatial) gradient at any given point. A screenshot of a sample program that can perform these calculations is shown in Figure 2.3.

### 2.3 Actuators

Based on the layout of the MFD, we are limited to actuation at the boundaries only. Assuming no chemical partitioning between the fluid in the channels and the gel scaffold, changing the channel concentration effectively provides boundary control. Actuators for this system should accordingly be chosen or designed to both mix certain concentrations of solution upstream of the gel region and deliver this concentration to the gel region boundaries. A wealth of microfluidic actuators exists to achieve these goals; a number of references can offer a comprehensive review of what currently exists [8, 18, 26, 32]. In what follows, we will discuss only one exemplary type of mixer and
Figure 2.3: Screenshot of a sample LabVIEW VI that reads grayscale images and extracts their intensity profiles.
one pump as a "simplest case" configuration for implementing feedback control.

2.3.1 Mixer

The small length scales of microfluidic devices lead to extremely low Reynolds number flows, which can make mixing challenging, since two streams brought together will tend to flow in laminate layers, relying on diffusion for mixing. (These are called T or Y mixers, due to their shape.) Complex channel geometries and/or the introduction of surface roughness can enhance mixing [26], and mixers with dead volumes as small as picoliters have been developed [16]. At the most basic level, though, T or Y mixers will serve the purpose, given long enough mixing chambers or slow enough flow rates, dictated by the Péclet number. If each channel contains a different fixed concentration of chemical, varying the ratio of flow rates in the two streams (holding the total flow rate constant) results in a change in the downstream, mixed concentration.

One way to implement this concept is to incorporate microvalves into the inlet streams that can open and close at various base frequencies and relative duty cycles [8, 32]. This was realized in a test configuration by fabricating a multi-layer device, made up of a flow layer and a control layer (Figure 2.4). The upper surface of the flow layer is enclosed by a thin, flexible PDMS membrane (20 – 30 μm) with the control layer stacked on top. Pressure in the flow layer (20 – 30 psi) causes the membrane to deflect downward, shutting off the channel.

The pressurizing and depressurizing of the control layers is controlled by LabVIEW and solenoid valves. A constant supply of pressurized air is supplied to each control channel and miniature, normally closed, 12V solenoid valve (Clippard P/N: E310C-1C012) allows this air to pressurize the control channel when activated. The activation of these solenoid valves, in turn, is controlled by signals sent from LabVIEW via a data acquisition (DAQ) card. The low-voltage signals from the DAQ are amplified to the required voltage through a simple MOSFET (Digikey P/N: IRF520NPBF-ND) and power supply circuit. This layout is shown in Figure 2.5.
Figure 2.4: Section of the proposed microvalve configuration showing the control layer (top) and the flow layer (bottom). The flexible membrane deflects downward when the control channel is pressurized and seals flow layer. (a) Solid model rendering (b) Cross-section view showing the stacking.

Figure 2.5: Configuration automated valve operation. 5V signals from a DAQ card activate a miniature solenoid valve which opens, allowing the control channel to be pressurized by the supply air and sealing the channel.
2.3.2 Pump

As with microfluidic mixers, the range of pumps in use for microfluidics spans simple to very complex. External volume-displacement pumps, such as syringe-pumps, can provide flow control down to the nano-liter per minute range [30], and can be computer-controlled. On-chip actuators, however, have advantages in their form factor and proximity to the actual plant. Again, the review by Iverson and Garimella [18] gives many examples of on-chip micropumps, including peristaltic, diaphragm, electroosmotic and thermal pumps. For their simplicity and similarity in design and use to the microvalves described in the previous section, we propose to use peristaltic pumps.

The operating principle of these pumps is straightforward: a number of control layers are stacked on top of a flow layer, as described for the microvalve above. These “valves” are then shut and open in a sequential fashion to cause fluid to displace (Figure 2.6). Each complete cycle pumps a fixed amount of fluid, governed by the width and spacing of the control channels, and the pumping rate can be changed by altering the frequency of opening and closing. The pressurization of the control channels is performed in exactly the same way as for the mixer valves described previously.

2.4 System limitations

The preceding sections of this chapter have described the three key components of a control system: plant, sensors, and actuators. The plant, which we take to be the gel-scaffold region within the MFD described in Sec. 2.1, is taken to be fixed. In reality, a better control system might be designed if the MFD were redesigned—for example, to allow for more actuation sites within the scaffold. Similarly, the choice of actuator as an upstream mixer and pump combination was made based on the existing device, when in fact the mixer and pump described are extremely limited.

Firstly, a purely diffusion-based mixer requires very long times and large volumes (large in the microfluidic sense) to ensure complete mixing. This causes significant
Figure 2.6: Proof of concept peristaltic pump. The pump is made up of three valves that open in close in a particular sequence to drive a volume flow. Shown here is an open valve (left) and a closed valve (right). Incomplete sealing is due to rectangular flow channels.
actuator delays, also limiting the rate of change of inputs. For example, based on a 1mm channel width, diffusive mixing times might be on the order of 10 minutes, and step changes in concentration would be nearly impossible. In this thesis, however, we focus on the mathematics of expressing this system in a way amenable to control, not the hardware itself. To this end, we will assume that the actuators are ideal or near-ideal and not plagued by the problems encountered in this first iteration of hardware. Given the breadth of mixers and pumps available for lab-on-chip assays [15, 18, 26, 32], this is not a particularly severe simplification.

Finally, while image-analysis is an information-rich and oft-used technique in quantifying biological experiments, for our control problem, it will be seen that this sensor only allows access to the system output, not system necessarily the internal states. This, of course, is a fairly common issue in real control systems, and much theory has been developed relating to observers and state estimation. Furthermore, because our control problem concerns only the concentration and gradient at a particular point, which the sensor gathers this information for the entire region, this excess data may be used to determine the internal states, although this is outside the scope of this work.
Chapter 3

Plant Description

3.1 Governing physics: chemical diffusion

Chemical diffusion is a common process for which the governing equation is well established. It is governed by Fick's laws and is identical to the heat diffusion equation if the coefficient of thermal diffusivity is replaced by the material diffusion coefficient, $D$. For chemical diffusion, $D$ is a function of both the solute (chemical species) and the solvent. The full 3D equation for a concentration of a single chemical $z(\vec{x}, t) = z(x, y, z, t)$ is

$$\frac{\partial z(\vec{x}, t)}{\partial t} = D \nabla^2 z(\vec{x}, t)$$  

(3.1)

which, in the 1D Cartesian coordinate system relevant to our MFD, reduces to

$$\frac{\partial z(x, t)}{\partial t} = D \frac{\partial^2 z(x, t)}{\partial x^2}$$  

(3.2)

Equation 3.2 is a parabolic partial differential equation (PDE) that is inherently stable. Instabilities may arise if production or reactions are considered, which will be discussed in Sec. 3.1.2.
3.1.1 Solutions to the PDE

Standard mathematical textbooks discuss the solution of partial differential equations. The diffusion equation (Eq. 3.1) can be approximated in 3D using finite element (FEM) methods [7]. Alternatively, analytical solutions can be found via separation of variables. The analytical solutions are dependent on all three of: geometry, boundary conditions, and initial conditions, and are infinite sums of sines and cosines [9]. In what follows we will inspect these solutions to give some insight into the resulting state-space properties.

For our plant, we consider “slab” geometry with Neumann (zero flux) boundary conditions. In other words, the Neumann boundary conditions are that $\partial z / \partial x = 0$ at $x = 0, x = L$. While this seems restrictive or possibly even counterintuitive initially, this boundary condition essentially matches the gel region boundary concentration to the channel concentration, which is indeed the desired result. Dirichlet boundary conditions ($z$ fixed at $x = 0, x = L$) would overconstrain the boundary concentrations.

The analytical solution for Neumann boundary conditions can be found by separation of variables and is given in [9]:

\[
\begin{align*}
  z(x, t) &= \frac{1}{L} \int_0^L f(x') \, dx' + \sum_{n=1}^{\infty} (a_n) \exp \left( -\frac{Dn^2\pi^2}{L^2} t \right) \cos \left( \frac{n\pi x}{L} \right) \\
  a_n &= \frac{2}{L} \int_0^L f(x') \cos \left( \frac{n\pi x'}{L} \right) \, dx'
\end{align*}
\]  

with $f(x)$ being the initial concentration distribution for $t < 0$. From the presence of this term in Eqs. 3.3 and 3.4 we can easily see the dependence of the solution on the initial conditions. More subtly, the boundary conditions are buried in the $\cos \left( \frac{n\pi x}{L} \right)$ term. Dirichlet conditions would yield sines as a solution, since the endpoints of sine functions are fixed at zero. As we expect, the solution is the product of a purely spatial function and a pure temporal function, with the temporal function decaying away exponentially. In the context of the control problem in question, the key observation to make in the analytical solution is that it is an infinite sum. It will
be shown later that this is an inherent characteristic of systems governed by PDEs and is related to the fact that they have an infinite number of internal states.

### 3.1.2 Role of reaction terms

Although chemical reactions will not be specifically addressed in this thesis, we will include a brief discussion of their potential role in the context of the control problem, as reactions might prove to be significant in the true angiogenic system under study. As mentioned previously, the diffusion equation itself (Eq. 3.2) is inherently stable. If we consider either production of chemical reactions between the chemical and the environment, however, this diffusion-reaction system may be driven unstable. If we consider an arbitrary local reaction (or production) term, $R$, the governing equation becomes

$$\frac{\partial z}{\partial t} = D \frac{\partial^2 z}{\partial x^2} + R$$

(3.5)

In the simplest case, where $R$ represents zeroth order production of the chemical such that $R$ is some constant greater than zero, Eq. 3.5 can clearly become unstable. In addition, $R$ would represent the effects of some binding reaction which could be first order irreversible

$$[A] \xrightarrow{k_1} [P]$$

$$R_A = -k_1[A]$$

second order irreversible,

$$[A] + [B] \xrightarrow{k_2} [P]$$

$$R_A = -k_2[A][B]$$
or, more interestingly, first and second order reversible,

\[
[A] \xrightleftharpoons[k^-]{k^+} [P] \quad \text{(first order)}
\]
\[
R_A = -k^+[A] + k^-[P]
\]

\[
[A] + [B] \xrightleftharpoons[k^-]{k^+} [P] \quad \text{(second order)}
\]
\[
R_A = -k^+[A][B] + k^-[P]
\]

These four equations describe the binding between \( A \) and \( B \) into some complex \( P \). In the irreversible cases, the reaction term is always less than zero and the system remains stable as before. In the reversible cases, however, if unbinding is more favorable than binding, \( R \) may be greater than zero and Eq. 3.5 may be driven unstable. In the context of angiogenesis, VEGF is both bound and produced by the endothelial cells [25]; thus, depending on specific reaction rates, is it possible that instabilities arise in the system.

### 3.2 The significance of a PDE in control systems

Partial differential equations arise in a number of classic mechanical engineering systems. As noted previously, the mass diffusion equation is identical to the heat diffusion equation for conduction through a medium. A slightly more complex problem is that of beam bending or a vibrating spring, which is governed by a hyperbolic PDE. These problems of controlling the temperature along the length of a conducting rod or the amplitude of vibration of a flexible beam are related to the problem addressed here.

The common feature of each of these systems is that they are distributed parameter systems (DPS), which by definition have an infinite-dimensional state-space [10]. Accordingly, transfer functions for this class of systems are infinite as well. As a result, DPS have a number of interesting properties not found in lumped parameter
(or finite) systems, namely: (1) DPS have an infinite number of poles and zeros (2) the choice of boundary conditions affects where the poles and zeros are located [10]. In practical implementation, an infinity of states is not manageable by traditional controller design and analysis techniques.

Parallels with these observations and with the separation of variable solution (Sec. 3.1.1) are encouraging. As Eq. 3.3 shows, the analytical solution is an infinite sum whose form is boundary condition-dependent. Thus, in the frequency domain, it follows that the system representation should be equally infinite and boundary condition-dependent.

A number of “traditional” techniques exist for manipulating DPS into systems that can be controlled by classical techniques. Generally, the infinite system is truncated to include enough orders for stability, and the finite approximation is used [4, 10]. Other methods include boundary control, which maps unstable systems onto stable ones with desired boundary conditions and requires no approximation [21], or more mathematical techniques, such as defining the state-transition matrix as the Laplacian operator \((A = \partial/\partial x)\) and writing Eq. 3.2 in state-space form [11]. These techniques will be described in more detail in Sec. 4.1.
THIS PAGE INTENTIONALLY LEFT BLANK
Chapter 4

Conversion of a PDE to a control system model

4.1 Prior work and existing methods

While lumped parameter models are clearly more tractable and convenient for use in control systems, continuum systems that require distributed parameter models are frequently encountered in engineering systems. Beam vibration, temperature control of a heat conducting rod and even certain population models are all distributed parameter systems are require special consideration from a control perspective \[11\]. There are many mathematical treatments of these problems; some approximate the system and then employ classical controller methods while others treat the infinite system but are much more mathematical in nature. In what follows we will briefly review some of the primary methods in the first category.

4.1.1 Literature survey

Publications regarding control of DPS date back to the early 1960s and span a wide class of problems. A broad overview of such systems is given in a textbook by Junkins and Kim \[20\], which discusses the practical issues of controlling flexible systems, particularly in the context of spacecraft control. The text highlights the major math-
mathematical treatments of DPS, both infinite-dimensional and approximate. The approximations focus on spatial discretization; namely, finite element methods (FEM) and modal decomposition. Controller options for these finite approximations are also discussed.

A second approximation method is to project the system onto an orthogonal space of eigenvectors. Although this coordinate system is also infinite dimensional, it can be approximated by including only a finite number of eigenvectors, based on the lowest eigenvalue that guarantees stability. This idea was presented first in Balas, 1979 and has been oft cited and employed since. This method requires that we first have an exact model of the system, and that the PDE is at least quasi-linear (i.e. linear in the highest order derivative) [17].

Finally, while it is typically more convenient in multi-input, multi-output (MIMO) systems to describe the systems in state-space directly, the plant can also be described via a matrix of transfer functions that relate each input to each output. In order to develop this formulation, it is necessary to understand how to derive the infinite-dimensional transfer function for a SISO system. A 2009 tutorial publication by Curtain [10] methodically outlines this process. Curtain also describes how to express the resulting transfer functions using their partial fraction expansion, which leads to an infinite sum of transfer functions that can then be truncated up to some summation N. Included in the paper are a wide range of sample applications, as well as some concluding remarks regarding controller design. Following is a brief description of each of the methods just described.

4.1.2 FEM and Modal Decomposition

Finite element approximation of continuous systems is commonly used to make the system computable in a practical sense. In the FEM approximation of DPS, the spatial component is discretized, while the problem remains continuous in time [20].

Modal decomposition, or the “assumed modes” method, also results in a spatial discretization. In this method, the coupled space- and time-dependent function is modeled as a finite series of functions only dependent on space multiplied by time-
dependent functions. The time functions describe the amplitude of each mode while the space functions become a basis for a new coordinate system. Their product is often referred to as a shape function [20]. This method is commonly used in beam vibration theory, where the distributed mass, damping, and stiffness properties can be assembled in approximate mass, damping and stiffness matrices.

4.1.3 Orthogonal Projection

As previously stated, this method involves projecting the system onto a new coordinate system of eigenvectors, $\phi_k, k = 1, 2, 3, \ldots$. These eigenvectors have corresponding eigenvalues $\lambda_k$ such that $\lambda_1 \leq \lambda_2 \leq \ldots$. The approximation is then achieved by using two projection matrices such that each projects one subset of eigenvectors, corresponding with either the fast or slow eigenvalues. The system is then taken only to be the projection of the first $N$ eigenvalues. Without true mathematical rigor, this is described as follows.

Given a variable $z(x,t)$, assume that it can be expressed as the product of a space function and a time function,

$$ z(x,t) = \sum_{k=1}^{\infty} z_k(t)\phi_k(x), $$

where $\phi_k$ are the orthogonal basis functions. For the linear diffusion PDE (Eq. 3.2), $\phi_k$ are the eigenvectors of the Laplacian operator $\partial^2 / \partial x^2$ [4]. We can define two projection operators, $P$ and $Q$,

$$ Pz = \sum_{k=1}^{N} z_k(t)\phi_k(x), $$

$$ Qz = \sum_{k=N+1}^{\infty} z_k(t)\phi_k(x) $$

such that $P$ projects the first $N$ modes and $Q$ projects the remainder. If $N$ is chosen to be large enough such that $\lambda_N$ is greater than a required decay rate, the approximate
system described by $Pz$ will be stable and we can design a controller for this system that will also control the full distributed system.

### 4.1.4 Infinite transfer functions

The infinite transfer function approach is useful when the input is directly related to the system variable ($z(x,t)$ or $\partial z(x,t)/\partial x$, for example) and the output is the system variable at some position of interest ($z(x_o,t)$). For example, in the case of temperature control of a conducting rod, the input might be either temperature or heat flux at the ends, with the output being temperature at some point along the rod. Since the heat equation and diffusion equation are equivalent, we can use the same setup for the problem at hand.

The key to developing the infinite transfer function is to take the Laplace transform of the original PDE with respect to time only. This leaves a full ordinary differential equation (ODE) in space, with the coefficients involving the Laplace $s$ variable. This ODE can then be solved analytically for $Z(x,s)$ by well-known techniques, where the upper case letter indicates that the time variable has been Laplace transformed. The solution will clearly be a function of the boundary conditions, as expected. In this derivation, however, one of the boundary conditions is an input as well and the output is $Z(x_o,s)$, such that the transfer function $G(s) = \frac{Z(x_o,s)}{U(s)}$ can be easily found algebraically. This perhaps verbose description will be made clearer in Sec. 4.3.1, in which the full derivation is shown.

### 4.2 Choice of methods

To represent this control system we have chosen the infinite transfer function method just described. While a state-space approach is often preferred for MIMO systems, this system lacks intuitive internal states upon which to build such a model, and lends itself more naturally to a transfer function description. The transfer function approach has further advantages as far as clearly illustrating the effect of each input on each output. It is also relatively straightforward to compute using standard programming
software, for example, while FEM code often requires additional software packages.

4.3 Resulting equations

Three mathematical steps are taken in order to transform the diffusion equation (Eq. 3.2) into a controller-ready form. First, the infinite-transfer function is derived from each input to each output. Since there are two inputs (concentration at each end of the gel region) and two outputs (concentration and gradient at a point in the gel region), the result is four transfer functions. Each of these transfer functions is then expressed in its partial fraction expansion—an infinite series—which is approximated by summing only up to some summation order, \( N \). Finally, these approximated transfer functions are assembled into a transfer function matrix which can be transformed into state-space form. It is this state-space description of the system that we design our controller around.

4.3.1 Infinite transfer functions

The aim in this section is to derive a transfer function from each system input to each system output and construct a matrix of transfer functions, \( G(s) \) that can be converted into state-space form. The key concept behind the infinite transfer function derivation is the use of the Laplace transform to convert the PDE into an ODE, with an \( s \)-variable coefficient. In our two input, two output system, \( G(s) \) will be a \( 2 \times 2 \) matrix, with each element being a transfer function from input \( i = 1, 2 \) to output \( j = 1, 2 \): \( G_{ji}(s) = \frac{Y_j}{U_i} \).

For convenience we reproduce the diffusion equation here:

\[
\frac{\partial z}{\partial t} = D \frac{\partial^2 z}{\partial x^2}
\]

Taking the Laplace transform with respect to time gives

\[
sZ(x, s) = D \frac{d^2 Z(x, s)}{dx^2}
\]

(4.1)
This is a second order, linear ODE:

$$\frac{d^2 Z(x, s)}{dx^2} - \frac{s}{D} Z(x, s) = 0$$ \hspace{1cm} (4.2)

which has the solution

$$Z(x, s) = A \sinh \left( \sqrt{\frac{s}{D}} x \right) + B \cosh \left( \sqrt{\frac{s}{D}} x \right)$$ \hspace{1cm} (4.3)

The inputs, \( u_1(t) = z(x = 0, t) \) and \( u_2(t) = z(x = L, t) \), and the outputs, \( y_1 = z(x_o, t) \) and \( y_2 = \partial z(x_o, t)/\partial x \), also have Laplace transforms:

$$U_1(s) = Z(x = 0, s), \quad U_2(s) = Z(x = L, s)$$ \hspace{1cm} (4.4)

$$Y_1(x_o, s) = Z(x_o, s), \quad Y_2(s) = dZ(x_o, s)/dx$$ \hspace{1cm} (4.5)

As can be seen by Eq. 4.4, the inputs define the boundary conditions that are used to solve Eq. 4.3. For example, the concentration profile resulting from manipulating the input at \( x = 0 \) \( (u_1) \) only would be found by solving Eq. 4.3 with the boundary conditions

$$Z(0, s) = U_1 \text{ and } Z(L, s) = 0$$ \hspace{1cm} (4.6)

Similarly, the boundary conditions associated with the \( x = L \) \( (u_2) \) input are

$$Z(0, s) = 0 \text{ and } Z(L, s) = U_2$$ \hspace{1cm} (4.7)

Clearly, since we have a linear ODE that describes the dynamics of each input-output pair, the responses to each input can be summed to find the total resulting profiles (concentration and gradient) that arises from both inputs being changed.

To generate each transfer function, then, we use the boundary conditions appropriate for each input to solve for the constants \( A \) and \( B \) in Eq. 4.3 above, which will result in \( Z(x, s) = f(s, x) U_i(s) \), where \( f(s, x) \) is some function of \( s \) and \( x \). Writing each \( Y_j \) in terms of \( Z(x, s) \), as in Eq. 4.5, we can easily divide through by each \( U_i(s) \) to find the transfer function. To illustrate this, the simplest example, the derivation
for the transfer function between input 1 and output 2, \( G_{12} \), is shown below.

Applying the boundary conditions for \( U_2 \) (Eq. 4.7) to the solution for \( Z(x, s) \) (Eq. 4.3) gives us the constants

\[
B = 0, \quad A = \frac{U_2(s)}{\sinh \left( \frac{s}{D} L \right)} \quad (4.8)
\]

Substituting these constants back into the solution gives

\[
Z(x, s) = \frac{\sinh \left( \frac{s}{D} x \right)}{\sinh \left( \frac{s}{D} L \right)} U_2(s) \quad (4.9)
\]

For \( Y_1 = Z(x_o, s) \), we simply evaluate Eq. 4.9 at \( x = x_o \) and divide through by \( U_2 \) to yield

\[
G_{12}(s) = \frac{Y_1(s)}{U_2(s)} = \frac{\sinh \left( \frac{s}{D} x_o \right)}{\sinh \left( \frac{s}{D} L \right)} \quad (4.10)
\]

Solving for the poles of the system, we use the characteristic equation \( \sinh \left( \frac{s}{D} L \right) = 0 \), which has a real root at \( \sqrt{\frac{s}{D}} L = 0 \). However, as a transfer function variable, \( s = i\omega \), and we must look at the complex roots, which are \( \sqrt{\frac{s}{D}} L = 0, \pi, 2\pi, \ldots \). An alternative way to interpret this would be to recognize that \( \sinh(i\omega) = isin(\omega) \); thus, due to the periodicity of sine functions, the transfer functions for this system have an infinite number of poles (and zeros, by the same argument). Furthermore, we know that for any matrix of transfer functions related to the same plant, all transfer functions will have the same denominator and therefore this is true for all of the transfer functions.

The derivations for the other transfer functions are slightly more complicated and are given in Appendix A. The transfer functions are:

\[
G_{11}(s) = \frac{\cosh \left( \frac{s}{D} x_o \right) \sinh \left( \frac{s}{D} L \right) - \sinh \left( \frac{s}{D} x_o \right) \cosh \left( \frac{s}{D} L \right)}{\sinh \left( \frac{s}{D} L \right)} \quad (4.11)
\]

\[
G_{21}(s) = -\sqrt{\frac{s}{D}} \frac{\cosh \left( \frac{s}{D} (x_o - L) \right)}{\sinh \left( \frac{s}{D} L \right)} \quad (4.12)
\]

\[
G_{22}(s) = \sqrt{\frac{s}{D}} \frac{\cosh \left( \frac{s}{D} x_o \right)}{\sinh \left( \frac{s}{D} L \right)} \quad (4.13)
\]
As expected, the transfer functions to \( y_2 = \partial z / \partial x \) have an additional \( \sqrt{\frac{z}{D}} \) term, that falls out of the derivative of the \( sinh \) and \( cosh \) terms in Eq. 4.3.

4.3.2 Partial fraction expansion

For implementation purposes, it is necessary to approximate Eqns. 4.11-4.13 to a finite dimension. This is done by writing the partial fraction expansion of each transfer function, which yields an infinite sum, and truncating the summation \([10]\). The partial fraction expansion can be found by computing the complex residues for each pole, such that the transfer function can be written as

\[
G(s) = \sum_{n=0}^{\infty} \frac{\text{res}(\lambda_n)}{s - \lambda_n},
\]

where \( \lambda_n \) is the \( n^{th} \) pole of the infinite transfer function, \( n = 0, 1, 2, \ldots \infty \). The numerator, \( \text{res}(\lambda_n) \) is the residue of the pole \( \lambda_n \). Mathematically speaking, the residue of any function \( G(s) \) at a pole \( \lambda \) is given as

\[
\text{res}(G, \lambda) = \lim_{s \to \lambda} (s - \lambda)G(s).
\]

For a function that can be expressed as the quotient of two functions, \( G(s) = \frac{N(s)}{D(s)} \), this simplifies to

\[
\text{res}(G, \lambda) = \frac{N(\lambda)}{D'(\lambda)}
\]

(4.14)

after applying L'Hospital's rule. Using this equation, it is straightforward, however tedious, to develop the partial fraction expansions for each transfer function by writing out the numerator and first derivative of the denominator for each pole. Again using the \( G_{12} \) transfer function as the simplest example, the poles are at \( s = -\frac{(\pi n)^2}{L^2} D \), from which we clearly see that the zeroth order residual (and thus the zeroth order of the transfer function expansion) is zero\(^1\). However, for the next pole, \( \lambda_1 = -\left(\frac{\pi}{L}\right)^2 D \), the

\[^{1}\text{It may be obvious to note that these poles } s = -\frac{(\pi n)^2}{L^2} D \text{ are also the poles of all of the transfer functions since all transfer functions have the same denominator.}\]
The numerator and derivative of the denominator are:

\[ N_{12}(\lambda_1) = i \sin \left( \frac{\pi \lambda}{L} \right) \quad D'_{12}(\lambda_1) = \frac{\mu}{2\pi D} \]

and the resulting residue is

\[ \text{res}_{12}(\lambda_1) = \frac{2\pi D}{L^2} \sin \left( \frac{\pi \lambda}{L} \right) \]

which allows us to write

\[ G_{12}(\lambda_1) = \frac{2\pi D \sin \left( \frac{\pi \lambda}{L} \right)}{L^2s + \pi^2 D} \]

Continuing in this fashion, calculating the numerator, denominator and residue for each \( \lambda_n \), we find that

\[ G_{12}(\lambda_n) = \frac{(-1)^{n+1}2(n\pi D)\sin \left( \frac{\pi \lambda}{L} \right)}{L^2s + (n\pi)^2 D} \quad (4.15) \]

\[ G_{12}(s) = \sum_{n=1}^{\infty} \frac{(-1)^{n+1}2(k\pi D)\sin \left( \frac{\pi \lambda}{L} \right)}{L^2s + (n\pi)^2 D} \quad (4.16) \]

\[ G_{12}(s) = \sum_{n=1}^{\infty} \frac{(-1)^{n+1}2(k\pi D)\sin \left( \frac{\pi \lambda}{L} \right)}{L^2s + (n\pi)^2 D} \quad (4.17) \]

The expansions for the remaining three transfer functions are

\[ G_{11} = \sum_{n=1}^{\infty} \frac{2(n\pi D)\sin \left( \frac{\pi \lambda}{L} \right)}{L^2s + (n\pi)^2 D} \quad (4.18) \]

\[ G_{21} = \sum_{n=1}^{\infty} \frac{2D(n\pi)^2\cos \left( \frac{\pi \lambda}{L} \right)}{L^2s + (n\pi)^2 D} \quad (4.19) \]

\[ G_{22} = \sum_{n=1}^{\infty} \frac{(-1)^{n+1}2D(n\pi)^2\cos \left( \frac{\pi \lambda}{L} \right)}{L^2s + (n\pi)^2 D} \quad (4.20) \]

The derivations of each of these can be found in Appendix B.

### 4.3.3 Approximation and state-space realization

The finite approximation is simple to create once the partial fraction expansions (Eqns. 4.17-4.20) have been obtained; simply end the summation at some \( n = N \).
Clearly, the choice of $N$ is a tradeoff between accuracy at higher $N$ and ease of computation with less summations. Stability can also become an issue when controllers are considered, as the finite approximations can have right-half plane zeros at higher $N$, causing a minimum-phase system to be approximated as non-minimum phase.

With the transfer functions in their finite form, all that remains is to construct a transfer function matrix

\[
\begin{bmatrix}
y_1 \\
y_2 \\
\end{bmatrix} =
\begin{bmatrix}
G_{11} & G_{12} \\
G_{21} & G_{22}
\end{bmatrix}
\begin{bmatrix}
u_1 \\
u_2
\end{bmatrix}
\]

and convert to state-space form. The transformation to a state-space representation from Eq. 4.21 is purely mathematical. Computational packages such as MATLAB are capable of doing these calculations for MIMO as well as SISO systems. The `sys` function in MATLAB can convert a matrix of transfer functions into a state-space system, and the `minimal` property will calculate the minimal realization\textsuperscript{2}. The result is a state-space system with $N$ states, i.e. equal to the number of summations:

\[
\begin{align*}
\dot{\chi} &= A\bar{\chi} + B\bar{u} \\
\bar{y} &= C\bar{\chi}
\end{align*}
\]

For the general system with $i$ inputs and $j$ outputs,

\[
\begin{align*}
\bar{\chi} &\in \mathbb{R}^{N\times 1} & \bar{y} &\in \mathbb{R}^{j\times 1} \\
A &\in \mathbb{R}^{N\times N} & C &\in \mathbb{R}^{j\times N} \\
B &\in \mathbb{R}^{N\times i}
\end{align*}
\]

(Note that here we use $\bar{\chi}$ as the state variable, rather than the traditional $\bar{x}$ to avoid confusion with the spatial variable.)

A drawback to this method is that the system states, which are sequential derivatives of the previous state, lose some of their physical value. However, the state-space

\textsuperscript{2}Many state-space textbooks also describe how to perform this calculation manually, and Pota (1996), provides a concise instructional document details this method for MIMO systems specifically [28].
formulation is convenient when designing controllers since it downplays the complications with the system being SISO vs. MIMO.
Chapter 5

Implementing feedback control

5.1 Description of control problem

Thus far, we have described both a one-dimensional diffusion system used to deliver chemical inputs to a cell-culture assay, as well as the dynamics of the concentration profile in response to changes in the endpoint concentration. It was shown in Chapter 3 that the diffusion equation is a distributed parameter system and that the transfer functions as well as the state space are infinite. In Chapter 4 we derived transfer functions between each of the inputs and outputs, found finite approximations for them, and converted the transfer function matrix into state-space form in order to cast the system in a form amenable to controller design. Now, in this chapter, we will address the feedback control portion.

The objective of feedback control here is to regulate the concentration and gradient of a single chemical within a one-dimensional diffusion system. This system is inherently stable with no disturbances, meaning that stabilization is not one of aims of this controller. Instead, we focus on improving the speed of response and tracking references. The inputs are an imposed concentration at the endpoints of the region. In mathematical terms, this can be stated in the following manner:

Given a state-space system governed by the PDE \( \frac{\partial z}{\partial t} = D \frac{\partial^2 z}{\partial x^2} \), we wish to create a control law of the form \( \tilde{u} = -K\tilde{X} + \tilde{r} \), such that we can achieve one or all of the following:
1. Reach desired concentrations and/or gradients faster than the open-loop, diffusive time scale.

2. Track references for the desired outputs
   \[ y_1(t) = z(x_o, t) \] and \[ y_2(t) = \frac{\partial z(x_o, t)}{\partial x}. \]

3. Achieve concentration/gradient combinations that are not achievable with open-loop control.

The available inputs are \( u_1(t) = z(x = 0, t) \) and \( u_2(t) = z(x = L, t) \), i.e. the concentration in the channels. This is a 2-input, 2-output system: \( \vec{u} \in \mathbb{R}^{2 \times 1}, \vec{y} \in \mathbb{R}^{2 \times 1}. \)

Notation to be used in the remainder of this document is given in the following table:

<table>
<thead>
<tr>
<th>Property</th>
<th>Variable</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>time</td>
<td>( t )</td>
<td>seconds</td>
</tr>
<tr>
<td>position</td>
<td>( x )</td>
<td>( \mu m )</td>
</tr>
<tr>
<td>concentration</td>
<td>( z(x, t) )</td>
<td>ng/ml</td>
</tr>
<tr>
<td>inputs</td>
<td>( u_i(t) )</td>
<td>ng/ml</td>
</tr>
<tr>
<td>output</td>
<td>( y_1(t) )</td>
<td>ng/ml</td>
</tr>
<tr>
<td>output</td>
<td>( y_2(t) )</td>
<td>ng/ml/\mu m</td>
</tr>
<tr>
<td>state variable(^1)</td>
<td>( \chi_n, n = 1, 2, \ldots, N )</td>
<td>-</td>
</tr>
</tbody>
</table>

5.2 Options for controller design

Each of the three controller goals just described calls for a different controller design approach, and here we will describe these methods as they relate to each requirement.

As the emphasis of this work is on the formulation of the model equations, this section will focus on classical techniques, placing them in the context of this problem.

5.2.1 Pole Placement and Reference Tracking

The first goal, tuning the speed of response, can be achieved via pole placement (proportional control). In this method, a desired characteristic polynomial is chosen based on required time constants and controller poles are placed such that the closed
loop characteristic equation matches this polynomial. These poles determine the controller gain matrix, $K$, in the control law $\ddot{u} = -K\dot{X}$ or $\ddot{u} = -K\dot{X} + \vec{r}$. This method will increase the speed of response but often at the expense of high controller gains.

To reduce or eliminate the steady-state error, integral control can be added. This is achieved in state-space by augmenting the state vector to include the integral of the error, $\chi_I$:

$$\begin{align*}
\chi_I &= \int t \, \bar{e} \, dt, \\
\bar{e} &= \vec{r} - \vec{y},
\end{align*}$$

such that

$$\dot{\chi}_I = -C\chi + \vec{r}$$

This is described in many basic control texts for SISO systems [27], in which the integrator pole is placed in the same manner as just described. The only difference in this case is that the gain matrix, $K$, now contains both the proportional and integral gains, $K = [K_p, -K_e]$. For the MIMO case, though, $\vec{y}, \vec{r},$ and $\bar{e}$ all vectors, as shown above, such that $K_e$ becomes a matrix $K_e$, and the resulting state equations become

$$\begin{align*}
\begin{bmatrix}
\dot{X} \\
\dot{\chi}_I
\end{bmatrix} &= \begin{bmatrix}
A - BK_p & BK_e \\
-C & 0
\end{bmatrix}
\begin{bmatrix}
X \\
\chi_I
\end{bmatrix} +
\begin{bmatrix}
0 \\
1
\end{bmatrix}
\vec{r}
\quad (5.1)
\end{align*}$$

$$\vec{y} = \begin{bmatrix} C & 0 \end{bmatrix}
\begin{bmatrix}
\chi \\
\chi_I
\end{bmatrix}
\quad (5.2)$$

### 5.2.2 Optimal Control

In order to reach certain concentration/gradient pairs, it may be necessary to employ optimal control, as desired concentrations and gradients might conflict. In addition, optimal control can penalize certain control efforts, such as a maximum concentration and rate-of-change of concentration. These penalties prevent control laws that are unachievable using the chosen actuators. A typical control law seeks to minimize
some cost function, for example
\[ J = \int_0^\infty [x^TQx + u^TRu] \, dt, \]

where \( Q \) and \( R \) are the weighting matrices, designed to favor or penalize certain states or inputs. These matrices must be square and invertible, \( Q \in \mathbb{R}^{N \times N}, R \in \mathbb{R}^{2 \times 2}, \) i.e. with dimensions that match \( \bar{x} \) and \( \bar{u} \) respectively. For the optimal control case, reference tracking can be incorporated by replacing \( \bar{x} \) with \( \bar{x} - \bar{r} \) in the cost function.

5.2.3 State estimation

As a caveat, it is worth mentioning that each of the control laws just described assumes access to the states, \( \bar{x} \). In reality, these states are not measurable as they do not each represent a physical quantity. Thus, some form of state estimation is required in order to implement these laws. In designing our controller, we will assume that such an estimator can be developed. An example would be the traditional Luenberger observer, which compares the estimated output \( \hat{y} = C\hat{x} \) with the true output \( y \). This error is then incorporated into the estimate for future states, such that the state equations, assuming no measurement or process noise become:

\[
\begin{align*}
\dot{\hat{x}} &= A\hat{x} + Bu + L(\hat{y} - C\hat{x}) \\
\dot{\hat{x}} &= A\hat{x} + Bu \\
u &= -K\hat{x} + r
\end{align*}
\]

For visual clarity, here and in subsequent discussion we drop the vector notation for \( x, u \) and \( r \), as well as the boldface matrix notation for \( A, B, K \) and \( L \). \( L \) is the observer gain matrix.
5.3 Choice of control law

As a proof-of-concept, we designed example controllers based on the pole-placement methods and augmented state methods described above. The system was approximated to a second order system so that convenient metrics such as settling time and overshoot could be specified, although any number of poles can be placed. The performance of these two control methods is shown in the following chapter.
Chapter 6

Results

6.1 Plant representations

Here we present the results of the finite-dimensional approximations described in Chapter 3. As a baseline for comparison, we use the numerical solution to the 1D diffusion equation (Eq. 3.2). This was calculated using the MATLAB PDE solver, \texttt{pdeval}. The geometry for the system was based on the microfluidic system described in Sec. 2.1. The numerical solution was calculated for the diffusion of FITC Dextran in collagen. Dextran is a fluorescent molecule that can be produced at comparable molecular weight to VEGF (40 kDa for Dextran and 38kDa for VEGF) and is used as a surrogate molecule to simulate the behavior of VEGF. The value used for the diffusion coefficient of Dextran in collagen was taken from Ramanujan et al. (2002), which gives experimentally determined values for the diffusion coefficient of various molecular weights of Dextran in collagen gels [29].

Each input-output pair was analyzed separately in response to a step input in one channel, while holding the other input at zero. (The concentration was expressed in normalized units such that all responses were based on unity step response.) For example, in examining the behavior of concentration to control inputs, we looked at the step response of $G_{11}$ and $G_{12}$ individually. The actual output, the measurable concentration, is the sum of these responses, as this approximation is a linear system.

For the numerical solution, this is calculated by solving the PDE with boundary
conditions equal to the desired input. In other words, to verify the unity step response of $u_1 = z(x = 0, t)$, the numerical solution was found with boundary conditions $z(x = 0, t) = 1, z(x = L, t) = 0$.

Physically, this is the equivalent of having zero concentration of Dextran (or VEGF) in a given channel for $t < 0$, then immediately flooding the channel with some concentration at time $t = 0$ and then maintaining this concentration. While a true step input is likely not realizable, changes in concentration that are very fast compared to the time scale of diffusion are possible.

### 6.1.1 Approximation of the infinite transfer function

The step response of each finitely-approximated transfer function (Eqs. 4.18-4.20) was simulated in MATLAB and compared to the numerical solution for several summation numbers $N$ and for $x_0 = 600\mu$m. There was good agreement between the approximated systems and the numerical solutions, with accuracy increasing for higher $N$, as expected. This is shown in Figures 6.1 and 6.2.

One observed problem with this approximation is that at certain $N$, the step response of gradient to inputs has an early impulse-like behavior that causes an offset in the subsequent response (Figure 6.3). This offset remains constant through the remainder of the response and (i.e. the simulation in fact yields the correct final value if the initial offset is subtracted; it does not grow or diminish). Possible causes for this behavior are unclear.

As a practical check, the summed response was also calculated. To verify the linearity of the system, we also added the responses to each a step input of magnitude 2 in Channel 1 ($x = 0$) and of magnitude 1 in Channel 2 ($x = L$), and plotted this against the numerical solution with the corresponding boundary conditions, $z(x = 0, t) = 2, z(x = L, t) = 1$. This response also matches well with the numerical solution (Figure 6.4). The model was also verified for other positions of interest, $x_o$, shown in Figure 6.5.

Another significant issue that was briefly mentioned earlier is that the finite approximation may result in the system being non-minimum phase. As an example, we
Figure 6.1: Open-loop step response of each transfer function (solid), compared to the numerical solutions (dashed), for $N = 2$ summations, with the output at $x_o=600\mu m$. 
Figure 6.2: Open-loop step response of each transfer function (solid), compared to the numerical solutions (dashed), for $N = 20$ summations, with the output at $x_o = 600\mu m$. 
Figure 6.3: Open-loop step response of each transfer function for $N = 6$ summations, showing an offset in the response of gradient to $u_2$. 

*Figure 6.3*
Figure 6.4: Comparison of the complete response to two inputs \((u_1 = 2, u_2 = 1)\). The solid lines are the approximated response \((N = 23)\) while the dashed lines are the numerical solutions.
Total Output 1: $z(x)$ from both inputs with $u_1=2$, $u_2=1$

Figure 6.5: Concentration profiles over time at various positions $x_o$. (a) $x_o=100\mu m$ (b) $x_o=400\mu m$ (c) $x_o=600\mu m$ (d) $x_o=900\mu m$ (e) $x_o=1100\mu m$. 
examine the transfer function between input 1 and output 1, $G_{11}$ (Eq. 4.11). The true zeros of this system are $z_n = -\left(\frac{\pi n}{x_0}\right)^2 D$, all on the real axis in the left-half-plane. The zeros of the approximated system, however, can be complex, depending on the polynomial resulting from the approximation. For instance, for $N = 6$ summations, the zeros are

$$
z_1 = -0.39 \times 10^{-3}
$$

$$
z_2 = -1.65 \times 10^{-3}
$$

$$
z_3 = -3.48 \times 10^{-3}
$$

$$
z_{4,5} = -(0.24 \pm 0.96) \times 10^{-3}
$$

i.e. some complex but all in the left-half-plane. On the other hand, with $N = 20$, some of the complex poles are in the right-half-plane. These are shown in Figure 6.6.

### 6.1.2 Expression in state-space form

The state-space equations for the system are derived directly from the transfer function matrix, as such, we should not expect any difference in the response from the results just given. It is still worthwhile, however, to check that the code and MATLAB commands perform the operations intended. Shown in Figure 6.7 is the step response calculated by MATLAB based on the state-space objects, rather than the transfer functions directly. They are identical to the responses shown in Figure 6.1 and it is safe to say that any controllers designed around the state-space objects are indeed operating on the intended system approximation.

### 6.2 Controllers

Controllers based on pole placement and integral control were designed to both speed up the system dynamics and track a reference step input. An $N = 2$ system was used to demonstrate the performance of both controllers. Using pole placement, we were able to manipulate the system dynamics (Figures 6.8-6.9). Pole locations were chosen in a fairly arbitrary and heuristic manner, but as a proof of concept we can see that
Figure 6.6: Exact zeros (squares) compared to the zeros of the approximated system (circles) for $N = 6$ and $N = 20$. For higher summation numbers, accuracy comes at the cost of a potentially non-minimum phase system.
Figure 6.7: Open-loop step response for $N = 2$, calculated using the state-space formulation rather than the transfer functions. The response is identical to Figure 6.1.
this is an effective method for manipulating dynamics. However, the steady-state error for the desired concentration is quite high, and the required control efforts take on non-physical values (negative concentrations). This is shown in Figure 6.10 for the second case.

To improve on this, integral control was added via an augmented state, as described in Sec. 5.2.1. In this way, we were able to achieve zero steady-state error in response to reference step inputs for concentration and gradient simultaneously. (Of course, the references are not of the same magnitude or even necessarily the same sign, since the units and desired outputs are not equal.) Shown in Figure 6.11 is the response to the step reference $\mathcal{F} = [1, -0.001]$, $t > 0$, i.e. the desired output is concentration = 1 a.u. and gradient = -0.001 a.u./μm at $x_o$ for all $t > 0$. Again, this controller was implemented with an $N = 2$ approximation. While these results are encouraging in that they prove that a zero steady state error with low model order is achievable in theory, inspection of the control signals again shows that non-physical controls (negative concentrations) are called for (Figure 6.12). Thus, the need for a more sophisticated control law, such as optimal control, is clear, but beyond the scope of this work.
Figure 6.8: Tracking of a step reference using proportional control $\bar{u} = -K\bar{x}$ via pole placement, showing that the system dynamics can be manipulated. The step input was unity for output 1 (concentration) and $10^{-3}$ for output 2 (gradient), and poles were placed at $p_{1,2} = 10^{-4} \pm 2 \times 10^{-4}i$. This performance is clearly not ideal due to the oscillations and steady-state error.
Figure 6.9: A second controller employing the pole placement method, with poles chosen to have a higher damping ratio \( p_{1,2} = -3 \times 10^{-4} \pm 10^{-4}i \) than the previous case. The references are the same as in the previous case. Oscillations have been eliminated, but steady-state error still remains.
Figure 6.10: Controller efforts (top) and errors for concentration (bottom left) and gradient (bottom right). The controller design calls for a non-physical input at $x = L$ and there is a large steady-state error.
Figure 6.11: Tracking of a step reference using integral control via the augmented states method. The proportional poles were placed at $p_{1,2} = -3 \times 10^{-4} \pm 10^{-4}i$, as in Fig. 6.9, with integrator poles chosen to be ten times faster than the real part of the poles, i.e. $p_{11,12} = -30$. The error goes to zero for both concentration and gradient.
Figure 6.12: Controller efforts (top) and errors for concentration (bottom left) and gradient (bottom right). The controller again calls for a non-physical input, this time at $x = 0$. 
Chapter 7

Conclusions

7.1 General comments

Presented in this work was a method for describing a 2-input, 2-output one-dimensional diffusion process—a distributed parameter system—in a finite manner suitable for control. The outputs considered were concentration and gradient at a particular point in the system, and the inputs were concentrations at the endpoints. Distributed parameter systems exist frequently both in nature and in control problems; for example, systems with delays, heat conduction problems, acoustic waves, and vibrating beams. Control of such systems is frequently addressed in literature but is often highly theoretical in nature and not practical for implementation. Our method, in contrast, expresses the system in a form that classical controllers can easily be applied to.

The method employed here was an extension of the concept of infinite dimensional transfer functions. The finite-dimensional approximation of these transfer functions is developed by expressing them as an infinite series using partial fraction expansion and then truncating the series. For systems that are stable in their full form, the approximation will have a finite error relative to the exact system that decreases with increasing summations [10]. A serious consequence of this approximation, however, is that the zeros of the approximation are not the same as the original system and may even have right-half-plane zeros. Thus, a system that was once minimum phase may be approximated to a non-minimum phase system. This was seen in Figure 6.6.
While this is not ideal, the effect can be mitigated by the appropriate controller. Thus, the infinite transfer function method was extended to MIMO systems and was shown to adequately represent the true system.

7.2 Finite system approximation

As shown in Section 6.1.1 (Figs. 6.1-6.2), the finite dimensional approximation of each of four transfer functions represents the true system well, and more accurately with increasing $N$. One observed discrepancy was that at certain $N$, the gradient response to one input exhibited an impulse-like offset at early times (that remained constant for later times). The cause of this offset was unclear but did not cause the approximation overall output to be affected drastically; furthermore, with the appropriate application of feedback control, this proved to be inconsequential. Finally, the transfer function matrix was converted to a state-space system using MATLAB, and the response was verified for this as well.

7.3 Controller design

While the primary purpose of this work was to develop a set of state-space equations for the MIMO diffusion system, some preliminary controllers were also developed in order to assess whether these equations were appropriate. It was shown that with the derived equations, classical techniques such as pole placement and internal model design could be applied to change the system behavior. It was further shown that these techniques could be applied on a very low model order system, reducing computation significantly. However, a limitation of these classical controllers is that no provision is made for limits on the inputs or conflicting requirements between the two outputs. To this end, optimal control would need to be explored. The second significant limitation is that all of the controllers discussed assumed full state-feedback. This is useful only as an initial check of the utility of the state space system developed; in a true implementation, only the outputs are measurable. Thus, these controllers
alone are insufficient for implementing in a real system.

7.4 Future work

The next significant step in this work is to clearly work out the controller design and test the controller with the hardware discussed in Sections 2.2-2.3.2. On-chip mixers and pumps combined with microscopy-based sensing make up a control system that is implementable in LabView. Modification of the microfluidic system itself is a possibility as well, as the original system was designed with open-loop inputs in mind.

From a control perspective, the major hurdles are:

1. Lack of state information

2. Conflicting control requirements between the two outputs

3. Practical limitations on the achievable inputs

In a future controller, the last two issues may be addressed with an optimal controller, which may be state-based or output based. The first issue is the most complex. It is possible that this can be addressed using a standard observer, as noted in Section 5.2.3. More interestingly, this issue might be resolved using excess sensor information, as the imaged-based sensing provides output data for the entire region, despite being interested in only the output at a point. Given that the outputs at each point are related to those at every other point, this additional knowledge might be used to infer the states.

The formulation of the diffusion PDE into a state-space system was the first step in developing a control system capable of manipulating the concentration and gradient of a chemical that is diffusing within a porous media. This system is significant within a greater control problem—the study of angiogenesis—as chemical concentration and gradient are inputs to this latter system. Now that this formulation has been achieved, the next steps are to design this controller, subject to the limitations of the actual system, and integrate this module into the biological assay.
Appendix A

Transfer Function Derivations

In Section 4.3.1, the infinite transfer function, $G_{12}$, between Input 2 and Output 1 (concentration at $x = L$ and concentration at $x_0$, respectively) was derived. In this Appendix we will give the derivations for the remaining three transfer functions to include Input 1 (concentration at $x = 0$) and Output 2 (spatial gradient at $x_0$), as follows:

<table>
<thead>
<tr>
<th>$G_{11}$</th>
<th>Input 1, Output 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>$G_{21}$</td>
<td>Input 1, Output 2</td>
</tr>
<tr>
<td>$G_{22}$</td>
<td>Input 2, Output 2</td>
</tr>
</tbody>
</table>

The methodology is the same as presented in Section 4.3.1, but with more involved algebra. In each case, we begin with the solution to spatial ODE that results from taking the Laplace transform of the full PDE with respect to time,

$$Z(x, s) = A \sinh \left( \sqrt{\frac{s}{D}} x \right) + B \cosh \left( \sqrt{\frac{s}{D}} x \right)$$  \hspace{1cm} (A.1)

A.1 Transfer function $G_{11}$

For $G_{11}$, the input is delivered at $x = 0$ and therefore the boundary conditions are

$$Z(0, s) = U_1(s), Z(L, s) = 0.$$
This gives us

\[ B = U_1(s) \]  \hspace{1cm} (A.2) 

and

\[ A = \frac{-U_1(s) \cosh \left( \frac{x}{D} L \right)}{\sinh \left( \frac{x}{D} L \right)} \]  \hspace{1cm} (A.3) 

Substituting Eq. A.2 and Eq. A.3 back into the original ODE gives

\[ Z(x, s) = U_1(s) \cosh \left( \frac{x}{D} x \right) \sinh \left( \frac{x}{D} L \right) - \cosh \left( \frac{x}{D} L \right) \sinh \left( \frac{x}{D} x \right) \]  \hspace{1cm} (A.4) 

The Laplace transform of the output of interest in this case is \( Z(x_o, s) \), which is simply Eq. A.4 evaluated at \( x = x_o \). Doing this and dividing through by \( U_1 \) gives us the transfer function \( G_{11}(s) = \frac{Z(x_o, s)}{U_1(s)} \).

\[ G_{11}(s) = \frac{\cosh \left( \frac{x}{D} x_o \right) \sinh \left( \frac{x}{D} L \right) - \cosh \left( \frac{x}{D} L \right) \sinh \left( \frac{x}{D} x_o \right)}{\sinh \left( \frac{x}{D} L \right)} \]  \hspace{1cm} (A.5) 

## A.2 Transfer function \( G_{21} \)

For \( G_{21} \), the input and therefore the boundary conditions are identical to those for \( G_{21} \):

\[ Z(0, s) = U_1(s), Z(L, s) = 0. \]

Again, this gives us the same constants (Eqs. A.2-A.3) and the same resulting analytical solution for \( Z(x, s) \) (Eq. A.4). The difference in this case is that the output is now \( dZ(x_o, s)/dx \) (in the Laplace domain). Thus we must take the derivative, \( d/dx \) of Equation A.4 prior to dividing through by \( U_1 \).
\[
\frac{dZ(x, s)}{dx} = U_1(s) \left( \frac{d \cosh \left( \frac{s}{D} x \right) \sinh \left( \frac{s}{D} L \right)}{dx} - \frac{d \cosh \left( \frac{s}{D} L \right) \sinh \left( \frac{s}{D} x \right)}{dx} \right)
\]

\[
= U_1(s) \left( \frac{\sqrt{s}}{D} \sinh \left( \frac{s}{D} x \right) \sinh \left( \frac{s}{D} L \right) \right) - \frac{\sqrt{s}}{D} \cosh \left( \frac{s}{D} L \right) \cosh \left( \frac{s}{D} x \right) \right)
\]

Finally, dividing this result through by \( U_1 \) and evaluating at \( x_0 \) gives us

\[
G_{21}(s) = \frac{\frac{dZ(x_0, s)}{dx}}{U_1(s)}.
\]

\[
G_{21}(s) = \sqrt{\frac{s}{D}} \left[ \frac{\sinh \left( \frac{s}{D} x_0 \right) \sinh \left( \frac{s}{D} L \right) - \cosh \left( \frac{s}{D} L \right) \cosh \left( \frac{s}{D} x_0 \right)}{\sinh \left( \frac{s}{D} L \right)} \right] \quad (A.6)
\]

**A.3 Transfer function \( G_{22} \)**

This final transfer function is a response to input 2, which occurs at \( x = L \), and thus has different boundary conditions than the previous two cases. The boundary conditions for this case are

\[
Z(0, s) = 0, Z(L, s) = U_2(s).
\]

The resulting constants are

\[
B = 0 \quad (A.7)
\]

and

\[
A = \frac{U_2(s)}{\sinh \left( \frac{s}{D} L \right)} \quad (A.8)
\]

This gives the analytical solution

\[
Z(x, s) = U_2(s) \frac{\cosh \left( \frac{s}{D} x \right)}{\sinh \left( \frac{s}{D} L \right)} \quad (A.9)
\]
As for the case of the $G_{21}$ transfer function, we take the spatial derivative of this, evaluate at $x_o$, and divide by $U_2$:

\[
\frac{dZ(x,s)}{dx} = \frac{U_2(s)\sqrt{\frac{x}{D}}\cosh \left( \sqrt{\frac{x}{D}}x \right)}{\sinh \left( \sqrt{\frac{x}{D}}L \right)} \tag{A.10}
\]

\[
G_{22}(s) = \frac{dZ(x_o,s)}{dx} \frac{dx}{U_2(s)} \tag{A.11}
\]

\[
G_{22}(s) = \frac{\sqrt{\frac{x}{D}}\cosh \left( \sqrt{\frac{x}{D}}x_o \right)}{\sinh \left( \sqrt{\frac{x}{D}}L \right)} \tag{A.12}
\]

Finally, we have all four of the infinite dimensional transfer functions required to describe the relationship between the two inputs and the two outputs. The finite approximations are derived via partial fraction expansion, described in Appendix B.
Appendix B

Partial Fraction Expansions

The transfer function just derived were approximated to their finite form via partial fraction expansion, as described in Section 4.3.2. In this method, the transfer function $G(s)$ is expressed as

$$G(s) = \sum_{n=0}^{\infty} \frac{\text{res}(\lambda_n)}{s - \lambda_n}$$  \hspace{1cm} (B.1)

where $\text{res}(\lambda_n)$ is the complex residual of the pole $\lambda_n$. Ultimately, for transfer functions that can be expressed as a numerator $N(s)$ and a denominator $D(s)$, this becomes

$$G(s) = \sum_{n=1}^{\infty} \frac{N(\lambda_n)}{D'(\lambda_n)}$$  \hspace{1cm} (B.2)

This was detailed in Section 4.3.2, where the expansion for the $G_{12}$ transfer function was derived. In what follows, we will derive the remaining three expansions.

B.1 Partial Fraction Expansion for $G_{11}$

In order to write this expansion, three components are required: (1) the poles $\lambda_n$ (2) the elements of the numerator, $N_{11}(\lambda_n)$ and (3) the elements of the first derivative of the denominator, $D'_{11}(\lambda_n)$. The task is made simpler by the fact that all of the infinite transfer functions have the same denominator, meaning that they also share components (1) and (3) just described, which were already derived in the text and
are reproduced below.

\[ \lambda_n = -\left(\frac{n\pi}{L}\right)^2 D \]  
\[ D'(\lambda_n) = \frac{(-1)^{n+1}iL^2}{n2\pi D} \]  

All that remains is to derive \( N_{11}(\lambda_n) \). From Appendix A, we have that

\[ N_{11}(s) = \cosh\left(\sqrt{\frac{s}{D}x_o}\right) \sinh\left(\sqrt{\frac{s}{D}L}\right) - \cosh\left(\sqrt{\frac{s}{D}L}\right) \sinh\left(\sqrt{\frac{s}{D}x_o}\right) \]  

(B.5)

Substituting \( \lambda_n = -\left(\frac{n\pi}{L}\right)^2 D \) for \( s \) in this equation, the \( \sinh(\sqrt{\frac{s}{D}L}) \) terms are all zero and the \( \cosh(\sqrt{\frac{s}{D}L}) \) terms are identically equal to 1 such that

\[ N_{11}(\lambda_n) = (-1)^{n+1} \sin\left(\frac{n\pi x_o}{L}\right) \]  

(B.6)

Combining Eqs. B.4-B.6 gives the residual, \( \text{res}(\lambda_k) \), and with Eq. B.3, we have the complete expansion:

\[ G_{11}(s) = \sum_{n=1}^{\infty} \frac{2(n\pi D)\sin\left(\frac{n\pi x_o}{L}\right)}{L^2 s + (n\pi)^2 D} \]  

(B.7)

Note that we sum from \( n = 1 \) rather than from \( n = 0 \) as given in Eq. B.1, since it is clear from Eq. B.6 that the zeroth-order residual is 0.

### B.2 Partial Fraction Expansion for \( G_{21} \)

As previously noted, \( \lambda_n \) and \( D'(\lambda_n) \) are identical for all cases and given in Eqs. B.3-B.4. From Eq. A.6, we have that

\[ N_{21}(s) = \sqrt{\frac{s}{D}} \left( \sinh\left(\sqrt{\frac{s}{D}x_o}\right) \sinh\left(\sqrt{\frac{s}{D}L}\right) - \cosh\left(\sqrt{\frac{s}{D}L}\right) \cosh\left(\sqrt{\frac{s}{D}x_o}\right) \right) \]  

(B.8)
Substituting $\lambda_n$ for $s$ gives

$$N_{21}(\lambda_n) = (-1)^n \frac{n\pi}{L} \sin \left( \frac{n\pi x_o}{L} \right)$$  \hspace{1cm} (B.9)

This result, along with Eqs. B.3-B.4 allows us to write

$$G_{21} = \sum_{n=1}^{\infty} \frac{2D}{L} \frac{(n\pi)^2 \cos \left( \frac{n\pi x_o}{L} \right)}{L^2 s + (n\pi)^2 D}$$  \hspace{1cm} (B.10)

### B.3 Partial Fraction Expansion for $G_{22}$

Finally, we come to $G_{22}$. The numerator for $G_{22}$, from Eq. 4.13, is

$$N_{22}(s) = \sqrt{s} \cosh \left( \sqrt{\frac{s}{D}} x_o \right)$$  \hspace{1cm} (B.11)

which leads to

$$N_{22}(\lambda_n) = i \frac{n\pi}{L} \sin \left( \frac{n\pi x_o}{L} \right)$$  \hspace{1cm} (B.12)

Combining this, as before, with Eqs. B.3-B.4 gives

$$G_{22} = \sum_{n=1}^{\infty} \frac{(-1)^n (2D/L) (n\pi)^2 \cos \left( \frac{n\pi x_o}{L} \right)}{L^2 s + (n\pi)^2 D}$$  \hspace{1cm} (B.13)
Bibliography


