H_2-Optimal Sensor Location

by

Arman Tavakoli

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Applied Mathematics

Waterloo, Ontario, Canada, 2014

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Optimal sensor placement is an important problem with many applications; placing thermostats in rooms, installing pressure sensors in chemical columns or attaching vibration detection devices to structures are just a few of the examples. Frequently this placement problem is encountered while noise is present. The H_2-optimal control is a strategy designed for systems that have exogenous disturbing inputs. Therefore one approach for the optimal sensor location problem is to combine it with the H_2-optimal control. In this work the H_2-optimal control is explained and combined with the sensor placement problem to create the H_2-optimal sensor location problem.

The problem is examined for the one-dimensional beam equation and the two-dimensional diffusion equation in an L-shaped region. The optimal sensor location is calculated numerically for both models and multiple scenarios are considered where the location of the disturbance and the actuator are varied. The effect of different model parameters such as the weight of the state and the disturbance are investigated.

The results show that the optimal sensor location tends to be close to the disturbance location.

Acknowledgements

I am thankful to professors Kirsten Morris, Dong Eui Chang and Nathalie Lanson for teaching me mathematics.

I would also like to thank:

Prof. Michael Rubinstein for the happy face notation ☺(f(x)) vs (☺f) (x) and Pacman matrices;

Alinson Xavier for sharing math problems with me in the hallways, some took seconds to solve, some days and a few are still unsolved;

Robert Jonsson for refreshing discussions about math, physics and life.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Optimal sensor placement is an important problem encountered in many engineering projects and in daily life. A simple example would be the problem of placing a thermostat in a room. Suppose there is a window at one end and a heater at the other end. The room loses heat through the window and the heater needs to heat up the room. The thermostat is the temperature sensor. If the sensor is too close to the window, the heater will have to keep working until the area next to the window has reached the desired temperature. This can lead to the area closer to the heater becoming too warm or to higher heating costs because the heater was on for a long time. On the other hand if the sensor is too close to the heater, the heater would stop as soon as the area surrounding it has reached the desired temperature while the area closer to the window stays cold. A balance must be reached between the sensor position, the position of the heater and the window.

A cost function describes the relative weight of the different factors involved in the problem. The factors could include the cost of electricity, average room temperature and how much it deviates from the preset target in the thermostat. The cost function constitutes a sum of different factors, each with a weight. If comfort is the only important factor, a weight of one is assigned to the difference between the room temperature and the desired temperature while a weight of zero is assigned to all other factors. If electricity is expensive but at the same time a decently warm room is needed, both factors get a non-zero and comparable weight in the cost function. As its name suggests, the goal is to minimize the cost function. Using a cost function is a concrete way to compare different arrangements of the sensor, the heater and the window. The arrangement that gives the lowest cost is the best one.

Another important factor is the control law. The control law establishes the relationship between the sensor measurement and the action of the heater. The law describes the strategy that is used in the heater based on the thermostat measurement. In this work the control law is the H_2-optimal control. This control law is designed to deal with situations where there is disturbance or noise. It is assumed in H_2-control that all the noise or disturbance signals are White Gaussian (WG). In intuitive terms, a WG signal is unpredictable and as time goes on its past values are not related to its future values. A famous example for WG is the white screen seen on an analog TV when there is no signal. The TV makes a hissing sound and shows a random mixture of white and gray pixels. The cost function in H_2-optimal control is based on the statistical properties of the WG signals.

The problem of finding the optimal sensor location is also encountered in Active Vibration Control (AVC). In AVC, the goal is to reduce the vibration of a structure as much as possible. The vibrations in flexible structures die down naturally due to damping. However AVC can speed up this process. The location of the sensors and the actuators influence the settling time of the vibration. In Chapter 5 of this thesis the optimal sensor location for a simply supported beam is calculated.

This thesis is organized as follows. Chapter 2 consists of the background material which includes the definitions and the basic concepts in linear control theory. Chapter 3 introduces the H_2-optimal control and chapter 4 connects the H_2-optimal control with the optimal sensor location problem. A literature review is done which covers the application of H_2-optimal sensor placement to a variety of systems. In chapter 5 the optimal sensor location for the one-dimensional beam equation and the two-dimensional diffusion equation is calculated.

Chapter 2

Background

The introductory concepts related to the H_2-optimal control are reviewed here. These concepts can be found in numerous books such as [14], [23] and [31] and are included for reference and completeness.

2.1 Linear Time Invariant Systems

A Linear Time Invariant (LTI) system is defined as

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t), \qquad x(0) &= x_0, \\ y(t) &= Cx(t). \end{aligned} \tag{2.1}$$

The variables $x(t), u(t)$ and $y(t)$ are called the state, control and output and have sizes $n \times 1, m \times 1, p \times 1$ respectively. Matrices A, B and C are constant and have sizes $n \times n$, $n \times m$ and $p \times n$ respectively. Matrix B represents the actuator and C corresponds to the sensor.

Definition 2.1.1. For a square matrix A, define the **matrix exponential**

$$e^A = I + A + \frac{A^2}{2!} + \frac{A^3}{3!} + \dots \tag{2.2}$$

$$= \sum_{n=0}^{\infty} \frac{A^n}{n!}. \tag{2.3}$$

Theorem 2.1.1. If $x(t)$ satisfies equation (2.1) then,

$$x(t) = e^{At}x_0 + \int_0^t e^{A(t-\tau)}Bu(\tau)d\tau. \tag{2.4}$$

Proof. Using direct differentiation

$$\begin{aligned} \dot{x}(t) &= Ae^{At}x_0 + \int_0^t Ae^{A(t-\tau)}Bu(\tau)d\tau + e^{A(t-t)}Bu(t) \\ &= A\left(e^{At}x_0 + \int_0^t e^{A(t-\tau)}Bu(\tau)d\tau\right) + Bu(t) \\ &= Ax(t) + Bu(t). \end{aligned} \tag{2.5}$$

An alternative proof is to rewrite $\dot{x}(t) = Ax(t) + Bu(t)$ as

$$\begin{aligned} e^{-At}\dot{x}(t) - e^{-At}Ax(t) &= e^{-At}Bu(t) \\ \frac{d}{dt}(e^{-At}x(t)) &= e^{-At}Bu(t) \\ e^{-At}x(t) &= x(0) + \int_0^t e^{-A\tau}Bu(\tau)d\tau \\ x(t) &= e^{At}x_0 + e^{At}\int_0^t e^{-A\tau}Bu(\tau)d\tau. \end{aligned} \tag{2.6}$$

If $t = 0$,

$$x(0) = e^0x_0 + e^0\int_0^0 e^{-A\tau}Bu(\tau)d\tau, \tag{2.7}$$

$$x(0) = x_0. \tag{2.8}$$

Hence (2.4) satisfies the initial condition. □

A special case of (2.4) is when B is a column vector and $u(t) = \delta(t)$. In this case, (2.4) reduces to

$$x(t) = e^{At}(x_0 + B). \tag{2.9}$$

Another special case is $x(0) = x_0 + B$ and $u(t) = 0$. For this case (2.4) also reduces to (2.9). Therefore $(x(0) = x_0, u(t) = \delta(t))$ and $(x(0) = x_0 + B, u(t) = 0)$ lead to identical solutions.

A major topic in control theory is stability.

Definition 2.1.2. A square matrix A is called **Hurwitz** if all of its eigenvalues have negative real part.

Definition 2.1.3. The system (2.1) is **internally stable** if the matrix A is Hurwitz.

If $u(t) = 0$, (2.4) reduces to

$$x(t) = e^{At}x_0. \tag{2.10}$$

If A is Hurwitz then $\lim_{t\to\infty} |x(t)| = 0$ regardless of x_0. This is one motivation for Definition 2.1.3.

Two important concepts in control theory are stabilizability and detectability.

Definition 2.1.4. The pair (A, B) is called **stabilizable** if there exists a matrix K such that

$$A + BK \tag{2.11}$$

is Hurwitz.

If (A, B) is stabilizable, then it is possible to make (2.1) internally stable using the control $u(t)$. Choose the control $u(t) = Kx(t)$. Then (2.1) changes as

$$\begin{aligned} \dot{x} &= Ax + Bu \qquad (2.12)\\ &= Ax + BKx \qquad (2.13)\\ &= (A + BK)x. \qquad (2.14) \end{aligned}$$

If $A + BK$ is Hurwitz, the system is stable. This is the motivation for Definition 2.1.4. Since u depends on x, this choice of u is known as feedback control.

Definition 2.1.5. The pair (A, C) is called **detectable** if there exists a matrix L such that

$$A + LC \tag{2.15}$$

is Hurwitz.

An interesting problem is encountered when C in (2.1) is not invertible. For a system with a detectable pair (A, C) it is possible to reconstruct $x(t)$ using $y(t)$ and an auxiliary system called the *observer*. This is one motivation for Definition 2.1.5.

Theorem 2.1.2. (Luenberger Observer) Recall (2.1)

$$\begin{aligned}\dot{x}(t) &= Ax(t) + Bu(t), \quad x(0) = x_0,\\ y(t) &= Cx(t).\end{aligned}$$

Assume (A, C) is detectable. Let L be a matrix such that $A - LC$ is Hurwitz. Define the observer system

$$\begin{aligned}\dot{\hat{x}}(t) &= A\hat{x}(t) + L(y(t) - \hat{y}(t)) + Bu(t),\\ \hat{y}(t) &= C\hat{x}(t).\end{aligned} \tag{2.16}$$

Then regardless of the initial condition of the observer $\hat{x}(0)$

$$\lim_{t\to\infty} |x(t) - \hat{x}(t)| = 0. \tag{2.17}$$

Proof. Define $e(t) = x(t) - \hat{x}(t)$. Then

$$\begin{aligned}\dot{e}(t) &= \dot{x}(t) - \dot{\hat{x}}(t)\\ &= Ax(t) + Bu(t) - (A\hat{x}(t) + L(y(t) - \hat{y}(t)) + Bu(t))\\ &= (A - LC)(x(t) - \hat{x}(t))\\ &= (A - LC)e(t).\end{aligned} \tag{2.18}$$

Since $A - LC$ is Hurwitz, $\lim\limits_{t\to\infty} |e(t)| = 0$. □

Equation (2.1) can also be analyzed using the Laplace transform. Let $\mathbf{x}(s), \mathbf{u}(s)$ and $\mathbf{y}(s)$ be the Laplace transform of the variables $x(t), u(t)$ and $y(t)$ respectively. Then (2.1) is rewritten

$$s\mathbf{x}(s) - x(0) = A\mathbf{x}(s) + B\mathbf{u}(s), \tag{2.19}$$

$$\mathbf{x}(s) = (sI - A)^{-1}B\mathbf{u}(s) + (sI - A)^{-1}x(0), \tag{2.20}$$

$$\mathbf{y}(s) = C\mathbf{x}(s). \tag{2.21}$$

If $x(0) = 0$,

$$\mathbf{y}(s) = C(sI - A)^{-1}B\mathbf{u}(s). \tag{2.22}$$

Definition 2.1.6. The function $G(s) = C(sI - A)^{-1}B$ is called the **transfer function** between $u(s)$ and $y(s)$.

2.2 Linear Quadratic Regulator (LQR)

Linear-Quadratic Regulator (LQR) is an optimal control problem where the dynamical system is linear but the cost function is quadratic.

Lemma 2.2.1. [23, chapter 5, page 178] or [31, chapter 13, page 338]. Suppose (A, B) and (A, C) are stabilizable and detectable respectively. Then the Riccati equation

$$A^*X + XA - XBB^*X + C^*C = 0 \tag{2.23}$$

has a unique symmetric positive semi-definite solution X. Furthermore, the matrix

$$A - BB^*X \tag{2.24}$$

is Hurwitz.

Definition 2.2.1. Let X be the solution to (2.23) defined in Lemma 2.2.1. Then X is called the **stabilizing** solution.

Lemma 2.2.2. [31, chapter 2, page 37] Let R be a real positive definite matrix. Then
1. there exists a symmetric positive definite matrix $R^{1/2}$ such that $R = R^{1/2}R^{1/2}$,
2. the inverse R^{-1} exists,
3. there exists a symmetric positive definite matrix $R^{-1/2}$ such that $R^{-1} = R^{-1/2}R^{-1/2}$, and
4. $R^{-1/2}R^{1/2} = I$.

Lemma 2.2.3. Let A and B be defined as in Lemma 2.2.1 and R and $R^{-1/2}$ be defined as in Lemma 2.2.2. Define $B_{\text{new}} = B(R^{-1/2})$. Then (A, B_{new}) is stabilizable.

Proof. By Definition 2.1.4, there exists a matrix K such that $A + BK$ is Hurwitz. Define $M = R^{1/2}K$. Then

$$A + B_{\text{new}}M = A + B(R^{-1/2})R^{1/2}K \tag{2.25}$$
$$= A + BK. \tag{2.26}$$

Therefore $A + B_{\text{new}}M$ is Hurwitz and (A, B_{new}) is stabilizable. □

Lemma 2.2.4. Let $A, B, B_{\text{new}}, C, R$ be defined as in Lemmas 2.2.1 and 2.2.3. Then the Riccati equation

$$A^*X + XA - XBR^{-1}B^*X + C^*C = 0 \tag{2.27}$$

has a unique stabilizing solution.

Proof. Using B_{new}, equation (2.27) is rewritten,

$$
\begin{aligned}
0 &= A^*X + XA - XBR^{-1}B^*X + C^*C && (2.28)\\
&= A^*X + XA - XBR^{-1/2}R^{-1/2^*}B^*X + C^*C && (2.29)\\
&= A^*X + XA - XB_{\text{new}}B^*_{\text{new}}X + C^*C. && (2.30)
\end{aligned}
$$

By Lemma 2.2.3, (A, B_{new}) is stabilizable. By Lemma 2.2.1, equation (2.30) has a unique stabilizing solution. □

Theorem 2.2.5. (LQR) Let (A, B) and (A, C) be stabilizable and detectable respectively. Let R be a positive definite matrix. Let X be the stabilizing solution of (2.27). Consider the system

$$\dot{x} = Ax + Bu, \qquad x(0) = x_0. \tag{2.31}$$

Define the cost function

$$J = \int_0^\infty x^*C^*Cx + u^*Ru \ dt. \tag{2.32}$$

Then
1. the control u_{opt} that minimizes (2.32) is $u_{\text{opt}}(t) = -R^{-1}B^*Xx(t)$,
2. the control u_{opt} stabilizes (2.31), and
3. the minimum value of (2.32) obtained using $u_{\text{opt}}(t)$ is $J_{\text{opt}} = x_0^*Xx_0$.

The proof of the theorem can be found in [23, chapter 5, page 175] or [31, chapter 14, page 378]. For a proof using dynamic programming see [5] and for a proof using the Hamiltonian approach see [21, chapter 6, pages 181 and 189].

2.3 Generalized Plant

A generalized plant, as its name suggests, is a generalization of a plant. In a generalized plant, all the exogenous variables such as disturbance or noise are grouped together into a vector signal $w(t)$. All of the control inputs are grouped into a vector denoted by $u(t)$. Therefore the generalized plant has two vector inputs w and u. Similarly it has two outputs z and y, where z is a cost function and y is a measurement signal that provides information about the plant. A generalized plant in both open and closed-loop configurations is shown in Figure 2.1.

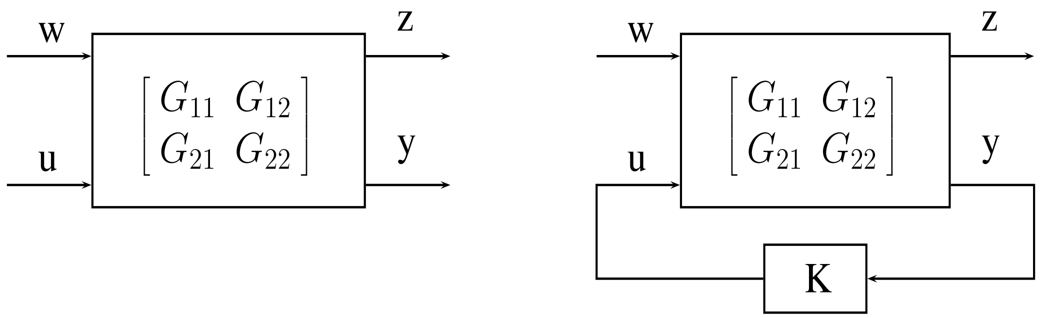

w
z
$\begin{bmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{bmatrix}$
u
y
w
z
$\begin{bmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{bmatrix}$
u
y
K

Figure 2.1: On the left there is a generalized plant in the open-loop configuration and on the right the system is closed-loop. The variable w represents the exogenous signals; u represents the control signals; z is the performance signal and it represents the cost that needs to be minimized. Variable y is the measurement output and provides information about the plant. In the closed-loop case the information in y is used to construct the control signal u. In the open-loop case there is no feedback and the control signal is created without y.

Mathematically, a generalized plant consists of four transfer functions, see Definition 2.1.6, from $(w, u) \rightarrow (z, y)$ respectively.

$$\begin{bmatrix} z(s) \\ y(s) \end{bmatrix} = \begin{bmatrix} G_{11}(s) & G_{12}(s) \\ G_{21}(s) & G_{22}(s) \end{bmatrix} \begin{bmatrix} w(s) \\ u(s) \end{bmatrix}. \tag{2.33}$$

If the system is closed-loop, the control signal is

$$u(s) = K(s)y(s). \tag{2.34}$$

Let T_{wz} be the closed-loop transfer function from w to z. From equations (2.33) and (2.34), T_{wz} can be calculated.

$$\begin{aligned} u &= Ky \\ &= K(G_{21}w + G_{22}u). \end{aligned} \tag{2.35}$$

In order for the system to be well-posed $(I - KG_{22})$ needs to be invertible. Equation (2.35) is rewritten as

$$u = (I - KG_{22})^{-1}KG_{21}w. \tag{2.36}$$

Using (2.33) and (2.36)

$$\begin{aligned} z &= G_{11}w + G_{12}u \\ &= (G_{11} + G_{12}(I - KG_{22})^{-1}KG_{21})w. \end{aligned} \tag{2.37}$$

Therefore

$$T_{wz} = G_{11} + G_{12}(I - KG_{22})^{-1}KG_{21}. \tag{2.38}$$

A similar expression for T_{wz} can be obtained using (2.33) and (2.34)

$$\begin{align} y &= G_{21}w + G_{22}u \tag{2.39}\\ &= G_{21}w + G_{22}Ky. \tag{2.40}\end{align}$$

Rewrite as

$$y = (I - G_{22}K)^{-1}G_{21}w. \tag{2.41}$$

Using (2.41), (2.34) and (2.33)

$$\begin{align} z &= G_{11}w + G_{12}u \tag{2.42}\\ &= G_{11}w + G_{12}Ky \tag{2.43}\\ &= G_{11}w + G_{12}K(I - G_{22}K)^{-1}G_{21}w. \tag{2.44}\end{align}$$

Hence

$$T_{wz} = G_{11} + G_{12}K(I - G_{22}K)^{-1}G_{21}. \tag{2.45}$$

Equations (2.38) and (2.45) are similar but not identical.

$$\begin{aligned} T_{wz} &= G_{11} + G_{12}(I - KG_{22})^{-1}KG_{21}\\ &= G_{11} + G_{12}K(I - G_{22}K)^{-1}G_{21}. \end{aligned} \tag{2.46}$$

Note that K appears in different places. This suggests the matrix identity

$$(I - KG_{22})^{-1}K = K(I - G_{22}K)^{-1}. \tag{2.47}$$

The next lemma establishes (2.47).

Lemma 2.3.1. Let $(I - KG_{22})$ be invertible. Then
1. $(I - G_{22}K)$ is invertible, and
2. $(I - KG_{22})^{-1}K = K(I - G_{22}K)^{-1}$.

Proof. 1. The contrapositive statement is: if $(I - G_{22}K)$ is not invertible then $(I - KG_{22})$ is not invertible. If $(I - G_{22}K)$ is not invertible then 1 is an eigenvalue of $G_{22}K$. Let x_1 be the corresponding eigenvector. Then $G_{22}Kx_1 = x_1$. Define $x_2 = Kx_1$. Then

$$\begin{aligned} KG_{22}x_2 &= KG_{22}Kx_1\\ &= Kx_1\\ &= x_2. \end{aligned} \tag{2.48}$$

Therefore x_2 is an eigenvector of KG_{22} with eigenvalue 1. Hence $(I - KG_{22})$ is not invertible.

2.

$$\begin{aligned} K - KG_{22}K &= K - KG_{22}K \\ K(I - G_{22}K) &= (I - KG_{22})K \\ (I - KG_{22})^{-1}K(I - G_{22}K) &= K \\ (I - KG_{22})^{-1}K &= K(I - G_{22}K)^{-1}. \quad \square \end{aligned} \tag{2.49}$$

An example is shown for calculating the closed-loop transfer function.

Example 1. Suppose

$$\begin{bmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{bmatrix} = \begin{bmatrix} \frac{1}{s+1} & \frac{1}{s+2} \\ \frac{1}{s+3} & \frac{1}{s+4} \end{bmatrix}, \tag{2.50}$$

$$K = \frac{1}{s+5}. \tag{2.51}$$

Then T_{wz} is calculated according to equation (2.38).

$$\begin{aligned} T_{wz} &= \frac{1}{s+1} + \frac{1}{s+2}\left(1 - \frac{1}{s+5} \cdot \frac{1}{s+4}\right)^{-1} \frac{1}{s+5} \cdot \frac{1}{s+3} \\ &= \frac{1}{s+1} + \frac{(s+5)(s+4)}{(s^2+9s+19)} \cdot \frac{1}{(s+5)(s+3)(s+2)} \\ &= \frac{s^4 + 14s^3 + 71s^2 + 154s + 118}{(s^2+9s+19)(s+3)(s+2)(s+1)} \\ &= \frac{s^4 + 14s^3 + 71s^2 + 154s + 118}{s^5 + 15s^4 + 84s^3 + 219s^2 + 263s + 114}. \end{aligned} \tag{2.52}$$

It is a natural question to ask how to go from a usual block diagram to a generalized plant. Consider the block diagram in Figure 2.2. Define

$$w = \begin{bmatrix} r \\ d \\ n \end{bmatrix}, \quad z = \begin{bmatrix} W_1(r - y) \\ W_2 u \end{bmatrix}. \tag{2.53}$$

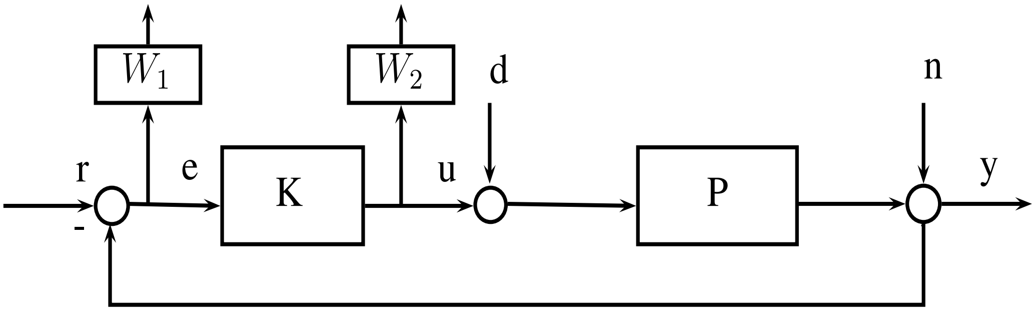


Figure 2.2: A block diagram with exogenous variables r, d, n and weights W_1, W_2. The controller K controls the plant P.

The output variable z and the measurement variable y need to be written only in terms of w and u. From the block diagram the following equations are obtained.

$$\begin{aligned} y &= n + P(d+u) \\ &= \begin{bmatrix} 0 & P & I \end{bmatrix} \begin{bmatrix} r \\ d \\ n \end{bmatrix} + Pu \qquad (2.54) \\ &= \begin{bmatrix} 0 & P & I \end{bmatrix} w + Pu. \end{aligned}$$

Using, (2.33) $G_{21} = \begin{bmatrix} 0 & P & I \end{bmatrix}$ and $G_{22} = P$. Similarly

$$\begin{aligned} z &= \begin{bmatrix} W_1(r-y) \\ W_2 u \end{bmatrix} = \begin{bmatrix} W_1 r - W_1 n - W_1 P d - W_1 P u \\ W_2 u \end{bmatrix} \\ &= \begin{bmatrix} W_1 & -W_1 P & -W_1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} r \\ d \\ n \end{bmatrix} + \begin{bmatrix} -W_1 P \\ W_2 \end{bmatrix} u \qquad (2.55) \\ &= \begin{bmatrix} W_1 & -W_1 P & -W_1 \\ 0 & 0 & 0 \end{bmatrix} w + \begin{bmatrix} -W_1 P \\ W_2 \end{bmatrix} u. \end{aligned}$$

Hence $G_{11} = \begin{bmatrix} W_1 & -W_1 P & -W_1 \\ 0 & 0 & 0 \end{bmatrix}$ and $G_{12} = \begin{bmatrix} -W_1 P \\ W_2 \end{bmatrix}$.

The generalized plant can also be described using state space equations. A generalized plant has a state vector x and a realization (A, B, C, D) such that

$$\begin{aligned} \dot{x} &= Ax + B_1 w + B_2 u, \qquad x(0) &= 0, \\ z &= C_1 x + D_{11} w + D_{12} u, & \qquad (2.56) \\ y &= C_2 x + D_{21} w + D_{22} u. \end{aligned}$$

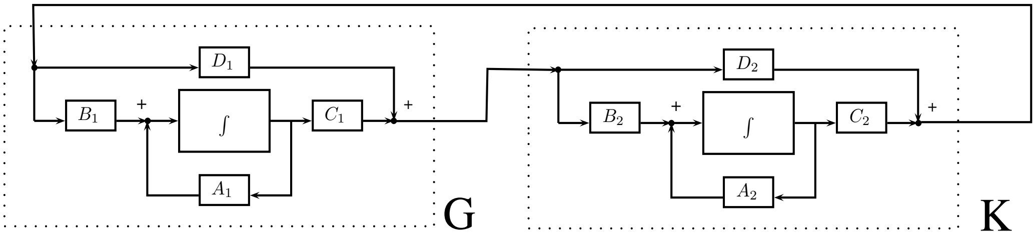


Figure 2.3: Two interconnected systems G and K create a larger system. The closed-loop connection of a plant and a controller is an example of such a connection.

A concise notation for the above equations is

$$\left[\begin{array}{c|cc} A & B_1 & B_2 \\ \hline C_1 & D_{11} & D_{12} \\ C_2 & D_{21} & D_{22} \end{array}\right]. \tag{2.57}$$

Given the state space equations, one can calculate the transfer functions $G_{ij}, i, j = 1, 2$ according to the formula $G_{ij} = C_i(sI - A)^{-1}B_j + D_{ij}$.

$$\begin{aligned} G_{11} &= C_1(sI - A)^{-1}B_1 + D_{11}, \\ G_{12} &= C_1(sI - A)^{-1}B_2 + D_{12}, \\ G_{21} &= C_2(sI - A)^{-1}B_1 + D_{21}, \\ G_{22} &= C_2(sI - A)^{-1}B_2 + D_{22}. \end{aligned} \tag{2.58}$$

Similarly given the transfer functions $G_{ij}, i, j = 1, 2$, a realization such as (2.57) can be found for the generalized plant. The procedure can be found in [31, chapter 3, page 69].

It is a common requirement for the controller to stabilize the plant. Two interconnected systems are part of a combined system that is larger than both of them (Figure 2.3).

Definition 2.3.1. A controller stabilizes a plant if the matrix A in the closed-loop system is Hurwitz.

Assume $A_1, B_1, C_1, D_1, x_1, y_1, u_1$ belong to G and $A_2, B_2, C_2, D_2, x_2, y_2, u_2$ belong to K. Then

$$\begin{aligned} \dot{x_1} &= A_1x_1 + B_1u_1, \quad y_1 = C_1x_1 + D_1u_1, \\ \dot{x_2} &= A_2x_2 + B_2u_2, \quad y_2 = C_2x_2 + D_2u_2, \\ u_1 &= y_2, \quad u_2 = y_1. \end{aligned} \tag{2.59}$$

Equations $u_1 = y_2$ and $u_2 = y_1$ connect the systems. The combined system has its own internal dynamics. Using (2.59)

$$u_1 = (I - D_2D_1)^{-1}(D_2C_1x_1 + C_2x_2), \tag{2.60}$$

$$u_2 = (I - D_1D_2)^{-1}(C_1x_1 + D_1C_2x_2). \tag{2.61}$$

Existence of $(I - D_2D_1)^{-1}$ is required for well-posedness. Using (2.59), (2.60) and (2.61), $\tilde{A}$ is found such that

$$\begin{bmatrix} \dot{x_1} \\ \dot{x_2} \end{bmatrix} = \tilde{A} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \tag{2.62}$$

$$\begin{bmatrix} \dot{x_1} \\ \dot{x_2} \end{bmatrix} = \begin{bmatrix} A_1 + B_1(I - D_2D_1)^{-1}D_2C_1 & B_1(I - D_2D_1)^{-1}C_2 \\ B_2(I - D_1D_2)^{-1}C_1 & A_2 + B_2(I - D_1D_2)^{-1}D_1C_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}. \tag{2.63}$$

Alternatively,

$$\tilde{A} = \begin{bmatrix} A_1 & 0 \\ 0 & A_2 \end{bmatrix} + \begin{bmatrix} B_1 & 0 \\ 0 & B_2 \end{bmatrix} \begin{bmatrix} I & -D_2 \\ -D_1 & I \end{bmatrix}^{-1} \begin{bmatrix} 0 & C_2 \\ C_1 & 0 \end{bmatrix}. \tag{2.64}$$

If $\tilde{A}$ is Hurwitz then controller K internally stabilizes G.

The next theorem discusses the state space equations for one stabilizing controller.

Theorem 2.3.2. [23, chapter 7, page 253], [31, chapter 5, page 121] Let

$$G = \left[\begin{array}{c|cc} A & B_1 & B_2 \\ \hline C_1 & 0 & D_{12} \\ C_2 & D_{21} & 0 \end{array}\right]$$

be a generalized plant. Assume (A, B_2) and (A, C_2) are stabilizable and detectable. Let F and L be matrices such that $A + B_2F, A + LC_2$ are Hurwitz. Then

$$K = \left[\begin{array}{c|c} A + B_2F + LC_2 & -L \\ \hline F & 0 \end{array}\right] \tag{2.65}$$

internally stabilizes G.

Proof. Using equation (2.64), $\tilde{A}$ of the combined system is.

$$\tilde{A} = \begin{bmatrix} A & 0 \\ 0 & A + B_2F + LC_2 \end{bmatrix} + \begin{bmatrix} B_2 & 0 \\ 0 & -L \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix}^{-1} \begin{bmatrix} 0 & F \\ C_2 & 0 \end{bmatrix} \tag{2.66}$$

$$= \begin{bmatrix} A & B_2F \\ -LC_2 & A + B_2F + LC_2 \end{bmatrix}. \tag{2.67}$$

Using a change of basis transformation $\tilde{A}$ is changed as

$$\begin{bmatrix} I & -I \\ 0 & I \end{bmatrix} \tilde{A} \begin{bmatrix} I & -I \\ 0 & I \end{bmatrix}^{-1} = \begin{bmatrix} I & -I \\ 0 & I \end{bmatrix} \begin{bmatrix} A & B_2F \\ -LC_2 & A + B_2F + LC_2 \end{bmatrix} \begin{bmatrix} I & I \\ 0 & I \end{bmatrix} \tag{2.68}$$

$$= \begin{bmatrix} A + LC_2 & 0 \\ -LC_2 & A + B_2F \end{bmatrix}. \tag{2.69}$$

Since (2.69) is lower triangular, the eigenvalues of (2.69) are the union of the eigenvalues of $A + LC_2$ and $A + B_2F$. Because both $A + LC_2$ and $A + B_2F$ are Hurwitz, (2.69) is Hurwitz. Since a change of basis transformation does not change the eigenvalues, $\tilde{A}$ has the same eigenvalues as (2.69). Therefore, $\tilde{A}$ is Hurwitz. □

2.4 Space RH_2

Let s be a complex variable and $p(s)$ and $q(s)$ be polynomials with real coefficients. Let deg(p) denote the degree of p. Define

$$S = \left\{ \frac{p(s)}{q(s)} \;\middle|\; \text{all roots of } q \text{ in the open left half plane}, \deg(p) + 1 \leq \deg(q) \right\}, \tag{2.70}$$

$$\tilde{S} = \left\{ \frac{p(s)}{q(s)} \;\middle|\; \text{all roots of } q \text{ in the open right half plane}, \deg(p) + 1 \leq \deg(q) \right\}. \tag{2.71}$$

Definition 2.4.1. Define RH_2 and $RH_2^{\sim}$ as the set of all matrices with elements in S and $\tilde{S}$ respectively.

Definition 2.4.2. A function $f(s)$ is **real-rational** if it can be written as $f(s) = \frac{p(s)}{q(s)}$, where $p(s)$ and $q(s)$ are polynomials with real coefficients.

Definition 2.4.3. Let $f(s) = \frac{p(s)}{q(s)}$ be a rational function. Then $f(s)$ is **proper** if $\deg(p) \leq \deg(q)$.

Definition 2.4.4. Let $f(s) = \frac{p(s)}{q(s)}$ be a rational function. Then $f(s)$ is **strictly proper** if $\deg(p) < \deg(q)$.

Definition 2.4.5. Let $f(s) = \frac{p(s)}{q(s)}$ be a proper rational function such that the greatest common divisor of $p(s)$ and $q(s)$ is 1. Then $f(s)$ is called **stable** if all the roots of $q(s)$ have negative real part.

If matrix A in Definition 2.1.6 is Hurwitz, the transfer function $G(s) = C(sI - A)^{-1}B$ is stable as in Definition 2.4.5 ([23, chapter 3, page 71]). This is the motivation for Definition 2.4.5.

The following result is well-known. For completeness, the proof is provided.

Lemma 2.4.1. Let $p_1(s)$ and $p_2(s)$ be polynomials such that $\deg(p_1) + 2 \leq \deg(p_2)$. Define $f(s) = \frac{p_1(s)}{p_2(s)}$. Then there exists a constant k such that

$$|f(s)| \leq \frac{k}{|s^2|} \qquad \text{as} \qquad |s| \to \infty. \tag{2.72}$$

Proof. Let $n = \deg(p_1)$, $m = \deg(p_2)$, $p_1(s) = \sum_{i=0}^{n} a_i s^i$ and $p_2(s) = \sum_{j=0}^{m} b_j s^j$. It is given that $n + 2 \leq m$. Since b_m is the leading coefficient in $p_2(s)$, $b_m \neq 0$. Choose

$$k = \frac{2}{|b_m|}(\sum_{i=0}^{n} |a_i|). \tag{2.73}$$

Using the following inequalities and (2.73), (2.72) is established. Using the Triangle inequality, for $1 \leq |s|$,

$$\begin{aligned}
|p_1(s)||s^2| &\leq \left(\sum_{i=0}^{n} |a_i|\right)(|s^n|)(|s^2|) && (2.74)\\
&\leq k\frac{|b_m|}{2}(|s^m|) && (2.75)\\
&= k\left(|b_m s^m| - |\frac{b_m}{2} s^m|\right). && (2.76)
\end{aligned}$$

Let $|s|$ be large enough such that

$$\begin{aligned}
\left(\sum_{j=0}^{m-1} |b_j|\right)\left(\frac{2}{|b_m|}\right) &\leq |s| && (2.77)\\
\left(\sum_{j=0}^{m-1} |b_j|\right) &\leq \left(\frac{|b_m|}{2}\right)|s|. && (2.78)
\end{aligned}$$

Using (2.78), (2.76) is continued as

$$k\left(|b_m s^m| - |\frac{b_m}{2} s^m|\right) \leq k\left(|b_m s^m| - \left(\sum_{j=0}^{m-1} |b_j|\right)|s^{m-1}|\right). \tag{2.79}$$

Using $1 \leq |s|$ and the Triangle inequality

$$\begin{aligned}
&\leq k\left(|b_m s^m| - \sum_{j=0}^{m-1} |b_j s^j|\right) && (2.80)\\
&\leq k\left(\left|\sum_{j=0}^{m} b_j s^j\right|\right) && (2.81)\\
&= k|p_2(s)|. && (2.82)
\end{aligned}$$

Therefore

$$\begin{aligned}
|p_1(s)||s^2| &\leq k|p_2(s)| && (2.83)\\
\left|\frac{p_1(s)}{p_2(s)}\right| &\leq \frac{k}{|s^2|}.
\end{aligned}$$

□

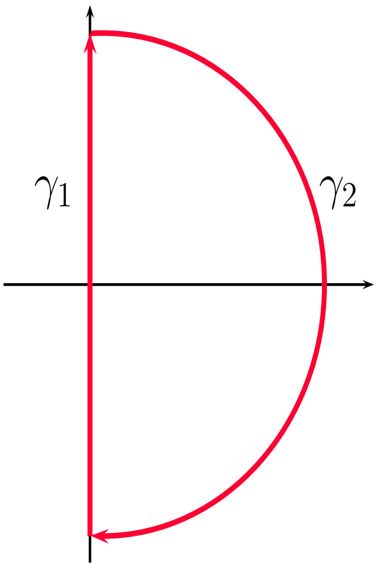


Figure 2.4: The contour used in Theorem (2.4.2).

Theorem 2.4.2. Let $f(s) \in S$ and $g(s) \in \tilde{S}$. Then

$$\int_{-\infty}^{\infty} f(iy)g^*(iy)dy = 0. \tag{2.84}$$

Proof. This result is stated without proof in [13, chapter 2, page 13] and [31, chapter 4, page 98]. Below, a proof using the Residue Theorem is provided.

Let R be the radius of the half circle and γ_1 be part of the contour that is on the imaginary axis in Figure 2.4. Then

$$\int_{-\infty}^{\infty} f(iy)g^*(iy)dy = \lim_{R\to\infty} \int_{-R}^{R} f(iy)g^*(iy)dy, \tag{2.85}$$

$$\int_{-R}^{R} f(iy)g^*(iy)dy = \frac{1}{i}\int_{\gamma_1} f(s)g^*(s)ds. \tag{2.86}$$

On γ_1, $s^* = -s$, $g^*(s) = g(-s)$ and

$$-i\int_{\gamma_1} f(s)g^*(s)ds = -i\int_{\gamma_1} f(s)g(-s)ds. \tag{2.87}$$

Define $h(s) = f(s)g(-s)$. Since $g(s)$ is in $\tilde{S}$, $g(-s)$ and $h(s)$ are in S. The function $h(s)$ has no poles on the imaginary axis and is meromorphic. Therefore the Residue Theorem can be applied to (2.87). For the definition of a meromorphic function and the Residue Theorem see [15]. Since $h(s) \in S$, it has no residues in the right half plane. Using the contour in Figure 2.4 and the Residue theorem

$$\int_{\gamma_1} h(s)ds + \int_{\gamma_2} h(s)ds = 0. \tag{2.88}$$

Let $h(s) = \dfrac{a(s)}{b(s)}$ where $a(s)$ and $b(s)$ are polynomials. Since $f(s), g(-s) \in S$, $2 + \deg(a) \leq \deg(b)$. Let k be the constant from Lemma 2.4.1. As $R \to \infty$

$$\begin{align}\left|\int_{\gamma_2} h(s)ds\right| &\leq \text{length}(\gamma_2) \cdot \sup_{\gamma_2} |h(s)| \tag{2.89}\\ &\leq \frac{1}{2} 2\pi R \frac{k}{R^2} \tag{2.90}\\ &= \frac{\pi k}{R}. \tag{2.91}\end{align}$$

Using (2.88) and (2.91),

$$\begin{align}\lim_{R\to\infty} \left|\int_{\gamma_1} h(s)ds\right| &= \lim_{R\to\infty} \left|\int_{\gamma_2} h(s)ds\right| \tag{2.92}\\ &\leq \lim_{R\to\infty} \frac{\pi k}{R} \tag{2.93}\\ &= 0. \tag{2.94}\end{align}$$

Using (2.85) and (2.92),

$$\begin{align}\left|\int_{-\infty}^{\infty} f(iy)g^*(iy)dy\right| &= \lim_{R\to\infty} |\int_{\gamma_1} h(s)ds| \tag{2.95}\\ &= 0. \qquad \square\end{align}$$

2.5 H_2-norm of a system

The H_2-norm of a transfer function is defined in this section. Consider the state space equations

$$\dot{x}(t) = A_1 x(t) + B_1 w(t) + B_2 u(t), \qquad x(0) = 0, \tag{2.96}$$
$$z(t) = Cx(t). \tag{2.97}$$

Assume the control $u(t)$ is in the feedback form $u(t) = Kx$. The gain K is determined based on the control objective; however in this section the focus is not on K and it is assumed that K has been determined. Equation (2.96) changes to

$$\dot{x}(t) = (A_1 + B_2 K)x(t) + B_1 w(t), \qquad x(0) = 0, \tag{2.98}$$
$$z(t) = Cx(t). \tag{2.99}$$

Define $A = A_1$ and $B = B_1$. Then (2.98) and (2.99) are rewritten as

$$\dot{x}(t) = Ax(t) + Bw(t), \qquad x(0) = 0, \tag{2.100}$$
$$z(t) = Cx(t). \tag{2.101}$$

Definition 2.5.1. The transfer function from w to z in the system (2.100)-(2.101) is

$$T_{wz}(s) = C(sI - A)^{-1}B. \tag{2.102}$$

The H_2-norm of T_{wz} is defined as

$$||T_{wz}||_{H_2} = \sqrt{\frac{1}{2\pi}\int_{-\infty}^{\infty} \operatorname{tr}\left(T_{wz}^*(iy)T_{wz}(iy)\right) dy}. \tag{2.103}$$

Two time-domain quantities that are equivalent to the H_2-norm (2.103) are mentioned below.

Definition 2.5.2. Let the size of $w(t)$ be $m \times 1$. Let $e_j, 1 \leq j \leq m$ be the standard basis for $\mathbb{R}^m$. Define $z_j(t)$ as

$$\dot{x}(t) = Ax(t) + Be_j\delta(t), \qquad x(0) = 0, \tag{2.104}$$
$$z_j(t) = Cx(t). \tag{2.105}$$

Lemma 2.5.1. Let A be a square matrix of size $n \times n$. Let $e_1, ..., e_n$ be the vectors in the standard basis where e_i has a one in the ith position and zero everywhere else. Then

$$\operatorname{tr}(A) = \sum_{i=1}^{n} e_i^* A e_i. \tag{2.106}$$

Proof. By definition

$$\begin{aligned} \operatorname{tr}(A) &= \sum_{i=1}^{n} A_{ii} \qquad (2.107) \\ &= \sum_{i=1}^{n} e_i^* A e_i. \qquad \square \end{aligned}$$

Theorem 2.5.2. [7, chapter 2, page 49] Let $z_j(t)$ and the H_2-norm of T_{wz} be defined as in Definition 2.5.2 and equation (2.103). Then

$$||T_{wz}||_{H_2}^2 = \sum_{j=1}^{m} \int_0^\infty z_j^*(t) z_j(t) dt. \tag{2.108}$$

Proof. Using equations (2.103), (2.102) and Lemma 2.5.1

$$\begin{aligned} ||T_{wz}||_{H_2}^2 &= \frac{1}{2\pi} \int_{-\infty}^{\infty} \operatorname{tr}(T_{wz}^*(jy) T_{wz}(jy))\, dy \qquad (2.109) \\ &= \sum_{i=1}^{m} \frac{1}{2\pi} \int_{-\infty}^{\infty} e_j^* B^* (iyI - A^*)^{-1} C^* C (iyI - A)^{-1} B e_j \, dy. \qquad (2.110) \end{aligned}$$

Let $\mathbf{z}_j(s)$ be the Laplace transform of z_j as defined in Definition 2.5.2. Using Definition 2.1.6

$$\mathbf{z}_j(s) = C(sI - A)^{-1} B e_j. \tag{2.111}$$

Using (2.111), (2.110) is rewritten as

$$||T_{wz}||_{H_2}^2 = \sum_{i=1}^{m} \frac{1}{2\pi} \int_{-\infty}^{\infty} \mathbf{z}_j(iy)^* \mathbf{z}_j(iy)\, dy. \tag{2.112}$$

Using Parseval's Theorem

$$||T_{wz}||_{H_2}^2 = \sum_{i=1}^{m} \int_0^\infty z_j(t)^* z_j(t)\, dt. \qquad \square$$

Definition 2.5.3. Define diag(a) as a diagonal matrix with a on all the diagonal elements.

Theorem 2.5.3. [7, chapter 2, page 50] Consider the system (2.100)-(2.101) with $x(0) = 0$. Let $w(t)$ be a $m \times 1$ column vector of white Gaussian noise signals such that

$$E[w(\tau_1)w(\tau_2)^*] = \text{diag}(\delta(\tau_1 - \tau_2)) \tag{2.113}$$

where E is the expectation operator. Then

$$||T_{wz}||_{H_2}^2 = \lim_{t\to\infty} E[z^*(t)z(t)]. \tag{2.114}$$

Proof. Using (2.4),

$$z(t) = \int_0^t Ce^{A(t-\tau)}Bw(\tau)d\tau. \tag{2.115}$$

Using (2.113) and (2.115)

$$E[z^*(t)z(t)] = E[\text{tr}(z(t)z^*(t)] \tag{2.116}$$

$$= E\left[\text{tr}\left(\left(\int_0^t Ce^{A(t-\tau_1)}Bw(\tau_1)d\tau_1\right)\left(\int_0^t Ce^{A(t-\tau_2)}Bw(\tau_2)d\tau_2\right)^*\right)\right] \tag{2.117}$$

$$= E\left[\text{tr}\left(\int_0^t\int_0^t Ce^{A(t-\tau_1)}Bw(\tau_1)w(\tau_2)^*B^*e^{A^*(t-\tau_2)}C^*d\tau_1d\tau_2\right)\right] \tag{2.118}$$

$$= \text{tr}\left(\int_0^t\int_0^t Ce^{A(t-\tau_1)}BE[w(\tau_1)w(\tau_2)^*]B^*e^{A^*(t-\tau_2)}C^*d\tau_1d\tau_2\right) \tag{2.119}$$

$$= \text{tr}\left(\int_0^t\int_0^t Ce^{A(t-\tau_1)}B\ \text{diag}(\delta(\tau_1-\tau_2))B^*e^{A^*(t-\tau_2)}C^*d\tau_1d\tau_2\right). \tag{2.120}$$

The diag($\delta(\tau_1 - \tau_2)$) term reduces the double integral to a single integral. The integration variables τ_1, τ_2 are replaced by one integration variable τ. Using Lemma 2.5.1

$$= \text{tr}\left(\int_0^t Ce^{A(t-\tau)}BB^*e^{A^*(t-\tau)}C^*d\tau\right) \tag{2.121}$$

$$= \text{tr}\left(\int_0^t (Ce^{A(t-\tau)}B)(Ce^{A(t-\tau)}B)^*d\tau\right). \tag{2.122}$$

For two matrices like X and Y, $\operatorname{tr}(XY) = \operatorname{tr}(YX)$. Therefore

$$= \operatorname{tr}\left(\int_0^t (Ce^{A(t-\tau)}B)^*(Ce^{A(t-\tau)}B)d\tau\right) \tag{2.123}$$

$$= \sum_{j=1}^{m} \int_0^t e_j^* B^* e^{A^*(t-\tau)} C^* C e^{A(t-\tau)} B e_j d\tau. \tag{2.124}$$

Using z_j as defined in Definition 2.5.2

$$= \sum_{j=1}^{m} \int_0^t z_j(\tau)^* z_j(\tau) d\tau. \tag{2.125}$$

Using (2.108) and (2.125)

$$\lim_{t\to\infty} E[z^*(t)z(t)] = \sum_{j=1}^{m} \int_0^\infty z_j(\tau)^* z_j(\tau) d\tau \tag{2.126}$$

$$= ||T_{wz}||_{H_2}^2. \qquad \square$$

Chapter 3

H_2-Optimal Control

The H_2-optimal control problem is about finding a stabilizing controller for a generalized plant that minimizes the H_2-norm of the closed-loop transfer function. The most general form of a generalized plant that is considered in this thesis has the following structure and is called the output feedback problem,

$$G = \left[\begin{array}{c|cc} A & B_1 & B_2 \\ \hline C_1 & 0 & D_{12} \\ C_2 & D_{21} & 0 \end{array}\right]. \tag{3.1}$$

For the state space equations of (3.1) see (2.56).

Definition 3.0.4. Let G be a generalized plant as in equation (3.1). Let K be any controller that stabilizes G, is real-rational, stable and proper as in Definitions 2.4.2, 2.4.5 and 2.4.3. The controller works in a feedback loop as in Figure 2.1. Let $\gamma(G, K)$ be the H_2-norm, as defined in (2.103), of the closed-loop system, as defined in (2.38). The optimal H_2-cost, $\gamma^\star$, is

$$\gamma^\star = \inf_K \gamma(G, K). \tag{3.2}$$

The H_2-optimal controller, $K^\star$, achieves the H_2-optimal cost, or in other words, $K^\star$ satisfies

$$\gamma^\star = \gamma(G, K^\star). \tag{3.3}$$

The solution of the output feedback problem is found by decomposing and reducing it into a sequence of more specialized and simpler problems. Specifically, four special problems are considered which are respectively state feedback, full information, full control and output estimation. Figure 3.1 shows the relationship between the different problems. In this chapter, each one of the special problems is solved independently, and the output feedback problem is solved by combining a full information and an output estimation problem. This approach is based on [31, chapters 12, 14]. A similar approach can be found in [11].

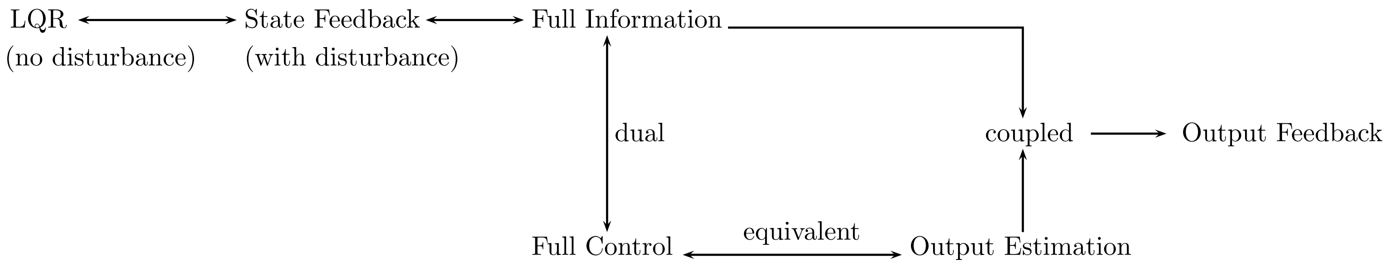


Figure 3.1: Output feedback problem is the most general problem. It can be decomposed into two simpler problems called full information and output estimation. The output estimation problem is equivalent to the full control problem which itself is dual to the full information problem. The sequence of simplifications from the general output feedback problem to the simpler full control and full information cases is the key step in solving the H_2-optimal control problem.

3.1 Assumptions

The following assumptions are imposed on the generalized plant.

- (A1) (A, B_2) is stabilizable.
- (A2) $D_{12}^* D_{12} = I$.
- (A3) $C_1^* D_{12} = 0$.
- (A4) $\begin{bmatrix} A - i\omega I \\ C_1 \end{bmatrix}$ has full column rank for all ω.
- (B1) (A, C_2) is detectable.
- (B2) $D_{21} D_{21}^* = I$.

- (B3) $B_1 D_{21}^* = 0$.
- (B4) $\begin{bmatrix} A - i\omega I & B_1 \end{bmatrix}$ has full row rank for all ω.

3.2 Special Problems

3.2.1 State Feedback

The state feedback generalized plant has the following structure.

$$G = \left[\begin{array}{c|cc} A & B_1 & B_2 \\ \hline C_1 & 0 & D_{12} \\ I & 0 & 0 \end{array}\right]. \tag{3.4}$$

The state feedback problem is similar to the LQR problem because in both of them the state variable is available for feedback, see [23, chapter 5, page 173]. However there are two main differences. First, in LQR the optimal cost depends on the initial condition while in the state feedback problem considered here the initial condition is set to zero. Second, in LQR there is no disturbance term while here there is a disturbance term $B_1 w(t)$ which affects the dynamics. Before the solution to the state feedback problem can be presented few algebraic lemmas are needed.

Lemma 3.2.1. [23, chapter 5, page 178] or [31, chapter 13, page 338]. Suppose (A, B) and (A, C) are stabilizable and detectable respectively. Then the Riccati equation

$$A^* X + XA - XBB^* X + C^* C = 0 \tag{3.5}$$

has a unique symmetric positive semi-definite solution X. Furthermore the matrix

$$A - BB^* X \tag{3.6}$$

is Hurwitz. The solution X with this property is called stabilizing.

Lemma 3.2.2. [31, chapter 13, theorem 13.7] If assumptions A1 and A4 from section 3.1 hold, the Riccati equation given below has a unique stabilizing and symmetric positive semi-definite solution.

$$A^* X + XA - XB_2 B_2^* X + C_1^* C_1 = 0. \tag{3.7}$$

Lemma 3.2.3. Let X be the unique stabilizing solution of the Riccati equation (3.7) and define A_F and C_F as

$$A_F = A - B_2 B_2^* X, \tag{3.8}$$
$$C_F = C_1 - D_{12} B_2^* X. \tag{3.9}$$

Then the following hold

1.

$$C_F^* D_{12} = -X B_2, \tag{3.10}$$

2.

$$A_F^* X + X A_F + C_F^* C_F = 0. \tag{3.11}$$

Proof. 1. Since X is symmetric $X = X^*$. Using (3.9)

$$\begin{aligned} C_F^* D_{12} &= (C_1 - D_{12} B_2^* X)^* D_{12} && (3.12)\\ &= (C_1^* - X^* B_2 D_{12}^*) D_{12} && (3.13)\\ &= C_1^* D_{12} - X B_2 D_{12}^* D_{12} && (3.14)\\ &= -X B_2. && (3.15) \end{aligned}$$

Assumptions A2 and A3 are needed for (3.14) and (3.15).

2. Again since X is symmetric $X = X^*$. Based on (3.7), the following equations hold.

$$\begin{aligned} 0 &= A^* X + X A - X B_2 B_2^* X + C_1^* C_1 && (3.16)\\ &= (A^* - X^* B_2 B_2^*) X + X(A - B_2 B_2^* X) + X B_2 B_2^* X + C_1^* C_1 && (3.17)\\ &= A_F^* X + X A_F + C_1^* C_1 + X B_2 D_{12}^* D_{12} B_2^* X \\ &\quad - C_1^* D_{12} B_2^* X - X^* B_2 D_{12}^* C_1 && (3.18)\\ &= A_F^* X + X A_F + (C_1 - D_{12} B_2^* X)^* (C_1 - D_{12} B_2^* X) && (3.19)\\ &= A_F^* X + X A_F + C_F^* C_F. && (3.20) \end{aligned}$$

Assumptions A2 and A3 are needed for (3.18). □

Definition 3.2.1. Define the following compact notation:

$$\left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right] = C(sI - A)^{-1} B + D. \tag{3.21}$$

Lemma 3.2.4. (Series Connection) Let $G_1 = \left[\begin{array}{c|c} A_1 & B_1 \\ \hline C_1 & D_1 \end{array}\right]$ and $G_2 = \left[\begin{array}{c|c} A_2 & B_2 \\ \hline C_2 & D_2 \end{array}\right]$ be connected in series. Then

$$G_1 G_2 = \left[\begin{array}{cc|c} A_1 & B_1 C_2 & B_1 D_2 \\ 0 & A_2 & B_2 \\ \hline C_1 & D_1 C_2 & D_1 D_2 \end{array}\right]. \tag{3.22}$$

Proof. Using definition 3.2.1,

$$\begin{aligned}
G_1 G_2 &= (C_1(sI - A_1)^{-1} B_1 + D_1)(C_2(sI - A_2)^{-1} B_2 + D_2) && (3.23)\\
&= C_1(sI - A_1)^{-1} B_1 C_2 (sI - A_2)^{-1} B_2 + C_1(sI - A_1)^{-1} B_1 D_2 && (3.24)\\
&+ D_1 C_2 (sI - A_2)^{-1} B_2 + D_1 D_2 && (3.25)\\
&= \left[\begin{array}{cc} C_1 & D_1 C_2 \end{array}\right] \left[\begin{array}{cc} (sI - A_1)^{-1} & (sI - A_1)^{-1} B_1 C_2 (sI - A_2)^{-1} \\ 0 & (sI - A_2)^{-1} \end{array}\right] \left[\begin{array}{c} B_1 D_2 \\ B_2 \end{array}\right] + D_1 D_2 && (3.26)\\
&= \left[\begin{array}{cc} C_1 & D_1 C_2 \end{array}\right] \left[\begin{array}{cc} (sI - A_1) & -B_1 C_2 \\ 0 & (sI - A_2) \end{array}\right]^{-1} \left[\begin{array}{c} B_1 D_2 \\ B_2 \end{array}\right] \\
&+ D_1 D_2 && (3.27)\\
&= \left[\begin{array}{cc} C_1 & D_1 C_2 \end{array}\right] (sI - \left[\begin{array}{cc} A_1 & B_1 C_2 \\ 0 & A_2 \end{array}\right])^{-1} \left[\begin{array}{c} B_1 D_2 \\ B_2 \end{array}\right] + D_1 D_2 && (3.28)\\
&= \left[\begin{array}{cc|c} A_1 & B_1 C_2 & B_1 D_2 \\ 0 & A_2 & B_2 \\ \hline C_1 & D_1 C_2 & D_1 D_2 \end{array}\right]. && (3.29)
\end{aligned}$$

To go from (3.26) to (3.27), use direct multiplication as

$$\begin{aligned}
&\left[\begin{array}{cc} (sI - A_1) & -B_1 C_2 \\ 0 & (sI - A_2) \end{array}\right] \left[\begin{array}{cc} (sI - A_1)^{-1} & (sI - A_1)^{-1} B_1 C_2 (sI - A_2)^{-1} \\ 0 & (sI - A_2)^{-1} \end{array}\right] && (3.30)\\
&= \left[\begin{array}{cc} (sI - A_1)(sI - A_1)^{-1} & B_1 C_2 (sI - A_2)^{-1} - B_1 C_2 (sI - A_2)^{-1} \\ 0 & (sI - A_2)(sI - A_2)^{-1} \end{array}\right] && (3.31)\\
&= \left[\begin{array}{cc} I & 0 \\ 0 & I \end{array}\right].
\end{aligned}$$

□

Definition 3.2.2. Let U be a matrix valued function of s. Define

$$U^{\sim}(s) = U^T(-s). \tag{3.32}$$

Note that U is transposed, not conjugate transposed.

Lemma 3.2.5. Let A_F and C_F be defined as (3.8), (3.9) and let U and $G_c(s)$ be defined as

$$U(s) = \left[\begin{array}{c|c} A_F & B_2 \\ \hline C_F & D_{12} \end{array}\right], \qquad G_c(s) = \left[\begin{array}{c|c} A_F & I \\ \hline C_F & 0 \end{array}\right]. \tag{3.33}$$

Then

1.

$$U^{\sim}U = I, \tag{3.34}$$

2.

$$U^{\sim}G_c \in RH_2^{\sim}. \tag{3.35}$$

Proof. 1. Using equation (3.32)

$$\begin{aligned} U^{\sim}(s) &= (C_F(-sI - A_F)^{-1}B_2 + D_{12})^T & (3.36) \\ &= B_2^T(sI - (-A_F)^T)^{-1}(-C_F)^T + D_{12}^T & (3.37) \\ &= \left[\begin{array}{c|c} -A_F^T & -C_F^T \\ \hline B_2 & D_{12}^T \end{array}\right]. & (3.38) \end{aligned}$$

It is best to either use transposition with symbol T or conjugate transposition with $*$. In most of this thesis $*$ is used, however definition (3.32) and equations (3.36), (3.37) and (3.38) specifically need transposition.

To calculate $U^{\sim}U$, two systems with appropriate transfer functions are connected in series as shown in Figure 3.2. The result can be derived either using transfer functions or state space equations. The transfer function solution follows from Lemma 3.2.4 by direction multiplication. For more insight, the state space solution is also presented.

Let x and p be the state variables and z, y the output variables of U and $U^{\sim}$ respectively. The output of U is the input of $U^{\sim}$.The state space equations of the two systems with output of U equal to the input of $U^{\sim}$ are

$$\begin{aligned} \dot{x} &= A_F x + B_2 u, & \dot{p} &= -A_F^* p - C_F^* z, \\ z &= C_F x + D_{12} u. & y &= B_2^* p + D_{12}^* z. \end{aligned} \tag{3.39}$$

Substitute z into (3.39) to arrive at:

$$\dot{p} = -A_F^* p - C_F^* C_F x - C_F^* D_{12} u, \qquad y = B_2^* p + D_{12}^* C_F x + D_{12}^* D_{12} u. \tag{3.40}$$

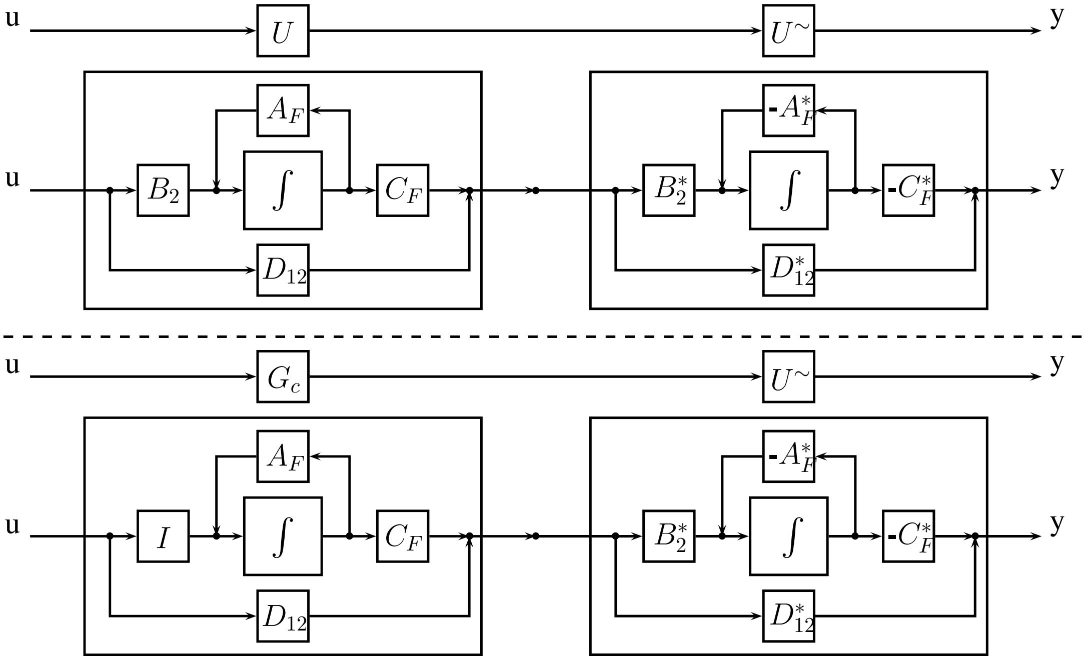


Figure 3.2: On top, there are two systems connected in series with transfer functions U and $U^{\sim}$. Similarly, in the bottom there are two systems in series with transfer functions G_c and $U^{\sim}$. Each system has a state space representation and the transfer function of the combined system can be found using the transfer functions or the state space equations.

Assumption A2 simplifies the last term on the right hand side to u. The state space equations for a system with joint state variable $\begin{bmatrix} p \\ x \end{bmatrix}$ are

$$\begin{bmatrix} \dot{p} \\ \dot{x} \end{bmatrix} = \begin{bmatrix} -A_F^* & -C_F^* C_F \\ 0 & A_F \end{bmatrix} \begin{bmatrix} p \\ x \end{bmatrix} + \begin{bmatrix} -C_F^* D_{12} \\ B_2 \end{bmatrix} u, \tag{3.41}$$

$$y = \begin{bmatrix} B_2^* & D_{12}^* C_F \end{bmatrix} \begin{bmatrix} p \\ x \end{bmatrix} + u. \tag{3.42}$$

Therefore

$$U^{\sim}U = \left[\begin{array}{cc|c} -A_F^* & -C_F^* C_F & -C_F^* D_{12} \\ 0 & A_F & B_2 \\ \hline B_2^* & D_{12}^* C_F & I \end{array}\right]. \tag{3.43}$$

Apply the state transformation $H = \begin{bmatrix} I & X \\ 0 & I \end{bmatrix}$. Under the transformation H, the state space system

$$\dot{x} = Ax + Bu \tag{3.44}$$

$$y = Cx + Du \tag{3.45}$$

transforms to

$$\dot{x}_{\text{new}} = H^{-1}AHx_{\text{new}} + H^{-1}Bu \tag{3.46}$$

$$y = CHx_{\text{new}} + Du \tag{3.47}$$

where $x_{\text{new}} = H^{-1}x$.

Note that $H^{-1} = \begin{bmatrix} I & -X \\ 0 & I \end{bmatrix}$. Using (3.10) and (3.11), components of $U^{\sim}U$ in (3.43) are changed as follows.

$$\begin{bmatrix} I & -X \\ 0 & I \end{bmatrix} \begin{bmatrix} -A_F^* & -C_F^* C_F \\ 0 & A_F \end{bmatrix} \begin{bmatrix} I & X \\ 0 & I \end{bmatrix} = \begin{bmatrix} I & -X \\ 0 & I \end{bmatrix} \begin{bmatrix} -A_F^* & -A_F^* X - C_F^* C_F \\ 0 & A_F \end{bmatrix} \tag{3.48}$$

$$= \begin{bmatrix} -A_F^* & -A_F^* X - XA_F - C_F^* C_F \\ 0 & A_F \end{bmatrix} \tag{3.49}$$

$$= \begin{bmatrix} -A_F^* & 0 \\ 0 & A_F \end{bmatrix}, \tag{3.50}$$

$$\begin{bmatrix} I & -X \\ 0 & I \end{bmatrix} \begin{bmatrix} -C_F^* D_{12} \\ B_2 \end{bmatrix} = \begin{bmatrix} -C_F^* D_{12} - XB_2 \\ B_2 \end{bmatrix} \tag{3.51}$$

$$= \begin{bmatrix} 0 \\ B_2 \end{bmatrix}, \tag{3.52}$$

$$\begin{bmatrix} B_2^* & D_{12}^* C_F \end{bmatrix} \begin{bmatrix} I & X \\ 0 & I \end{bmatrix} = \begin{bmatrix} B_2^* & B_2^* X + D_{12}^* C_F \end{bmatrix} \tag{3.53}$$

$$= \begin{bmatrix} B_2^* & 0 \end{bmatrix}. \tag{3.54}$$

Using (3.50), (3.52) and (3.54), Equation (3.43) is rewritten as

$$U^{\sim} U = \left[\begin{array}{cc|c} -A_F^* & 0 & 0 \\ 0 & A_F & B_2 \\ \hline B_2^* & 0 & I \end{array}\right] \tag{3.55}$$

$$= \begin{bmatrix} B_2^* & 0 \end{bmatrix} (sI - \begin{bmatrix} -A_F^* & 0 \\ 0 & A_F \end{bmatrix})^{-1} \begin{bmatrix} 0 \\ B_2 \end{bmatrix} + I \tag{3.56}$$

$$= \begin{bmatrix} B_2^* & 0 \end{bmatrix} \begin{bmatrix} (sI + A_F^*)^{-1} & 0 \\ 0 & (sI - A_F)^{-1} \end{bmatrix} \begin{bmatrix} 0 \\ B_2 \end{bmatrix} + I \tag{3.57}$$

$$= \begin{bmatrix} B_2^* & 0 \end{bmatrix} \begin{bmatrix} 0 \\ (sI - A_F)^{-1} B_2 \end{bmatrix} + I \tag{3.58}$$

$$= I. \tag{3.59}$$

2. Figure 3.2 shows two systems with transfer functions $U^{\sim}$ and G_c. Again, let x and p be the state variables and z, y be the output variables of G_c and $U^{\sim}$ respectively.

$$\begin{aligned} \dot{x} &= A_F x + u & \dot{p} &= -A_F^* p - C_F^* z & y &= B_2^* p + D_{12}^* z \\ z &= C_F x, & &= -A_F^* p - C_F^* C_F x, & &= B_2^* p + D_{12}^* C_F x. \end{aligned}$$

The joint state $\begin{bmatrix} p \\ x \end{bmatrix}$ satisfies the equations:

$$\begin{bmatrix} \dot{p} \\ \dot{x} \end{bmatrix} = \begin{bmatrix} -A_F^* & -C_F^* C_F \\ 0 & A_F \end{bmatrix} \begin{bmatrix} p \\ x \end{bmatrix} + \begin{bmatrix} 0 \\ I \end{bmatrix} u, \tag{3.60}$$

$$y = \begin{bmatrix} B_2^* & D_{12}^* C_F \end{bmatrix} \begin{bmatrix} p \\ x \end{bmatrix}. \tag{3.61}$$

Therefore $U^{\sim}G_c$ can be represented as:

$$U^{\sim}G_c = \left[\begin{array}{cc|c} -A_F^* & -C_F^*C_F & 0 \\ 0 & A_F & I \\ \hline B_2^* & D_{12}^*C_F & 0 \end{array}\right]. \tag{3.62}$$

Using (3.47), the components of $U^{\sim}G_c$ become

$$\begin{aligned} U^{\sim}G_c &= \left[\begin{array}{cc|c} -A_F^* & 0 & -X \\ 0 & A_F & I \\ \hline B_2^* & 0 & 0 \end{array}\right] && (3.63) \\ &= \left[\begin{array}{cc} B_2^* & 0 \end{array}\right](sI - \left[\begin{array}{cc} -A_F^* & 0 \\ 0 & A_F \end{array}\right])^{-1}\left[\begin{array}{c} -X \\ I \end{array}\right] && (3.64) \\ &= \left[\begin{array}{cc} B_2^* & 0 \end{array}\right]\left[\begin{array}{cc} (sI + A_F^*)^{-1} & 0 \\ 0 & (sI - A_F)^{-1} \end{array}\right]\left[\begin{array}{c} -X \\ I \end{array}\right] && (3.65) \\ &= \left[\begin{array}{cc} B_2^* & 0 \end{array}\right]\left[\begin{array}{cc} (sI + A_F^*)^{-1}(-X) & 0 \\ 0 & (sI - A_F)^{-1} \end{array}\right] && (3.66) \\ &= B_2^*(sI - (-A_F)^*)^{-1}(-X) && (3.67) \\ &= \left[\begin{array}{c|c} -A_F^* & -X \\ \hline B_2^* & 0 \end{array}\right]. && (3.68) \end{aligned}$$

Because X is a stabilizing solution for (3.7), A_F is Hurwitz. Therefore $-A_F^*$ has no poles in the left half plane. By Definition 2.4.1, $U^{\sim}G_c \in RH_2^{\sim}$. □

Lemma 3.2.6. Let $B_1, X, G_c(s)$ be defined by (3.4), (3.7) and (3.33) respectively. Then

$$||G_c(s)B_1||_{H_2}^2 = \mathrm{tr}(B_1^*XB_1). \tag{3.69}$$

Proof. Let the size of B_1 be $n \times m$ and let $e_1, ..., e_m$ be the standard basis of R^m. Consider the system below for $1 \leq j \leq m$,

$$\begin{aligned} \dot{x}(t) &= A_F x + B_1 e_j u(t), \qquad x(0) = 0, && (3.70) \\ z_j(t) &= C_F x(t). && (3.71) \end{aligned}$$

The transfer function is $\left[\begin{array}{c|c} A_F & B_1e_j \\ \hline C_F & 0 \end{array}\right]$, which is equal to $G_cB_1e_j$ because of (3.33). Choose $u(t) = \delta(t)$ and note that in the frequency domain $\boldsymbol{u}(s) = 1$. Based on Theorem 2.1.1

$$z_j(t) = C_F e^{A_F t} B_1 e_j \tag{3.72}$$

and in the frequency domain

$$\boldsymbol{z}_j(s) = G_c B_1 e_j. \tag{3.73}$$

Based on the definition of the H_2-norm, Lemma 2.5.1 and Parseval's Theorem,

$$\begin{aligned}
||G_c B_1||_{H_2}^2 &= \frac{1}{2\pi}\int_{-\infty}^{\infty} \operatorname{tr}(B_1^* G_c^*(i\omega) G_c(i\omega) B_1) d\omega & (3.74)\\
&= \frac{1}{2\pi}\int_{-\infty}^{\infty} \sum_{j=1}^{m} e_j^* B_1^* G_c^*(i\omega) G_c(i\omega) B_1 e_j d\omega & (3.75)\\
&= \frac{1}{2\pi}\int_{-\infty}^{\infty} \sum_{j=1}^{m} z_j^*(i\omega) z_j(i\omega) d\omega & (3.76)\\
&= \int_{0}^{\infty} \sum_{j=1}^{m} z_j(t)^* z_j(t) dt & (3.77)\\
&= \int_{0}^{\infty} \sum_{j=1}^{m} e_i^* B_1^* e^{A_F^* t} C_F^* C_F e^{A_F} B_1 e_i dt & (3.78)\\
&= \sum_{j=1}^{m} e_i^* B_1^* \int_{0}^{\infty} (e^{A_F^* t} C_F^* C_F e^{A_F} dt) B_1 e_i. & (3.79)
\end{aligned}$$

The integral term in Equation (3.79) is the solution to the equation (3.11) and is equal to X, see for example [31, chapter 3, page 71]. Using Lemma 2.5.1

$$||G_c B_1||_{H_2}^2 = \operatorname{tr}(B_1^* X B_1). \qquad \square$$

Lemma 3.2.7. Let G be defined by (3.4) and let $K(s)$ be a proper controller in the feedback configuration as in Definition 2.4.3 and (2.34). Then the transfer function from the disturbance w to the state x is

$$T_{wx} = (sI - A - B_2 K(s))^{-1} B_1. \tag{3.80}$$

Proof. Using (3.4) and (2.34)

$$\begin{aligned}
x(s) &= (sI - A)^{-1} B_1 w(s) + (sI - A)^{-1} B_2 u(s), & (3.81)\\
u(s) &= K(s) y(s). & (3.82)
\end{aligned}$$

In the state feedback problem $y(s) = x(s)$. Using (3.81) and (3.82)

$$\begin{aligned} x(s) &= (sI-A)^{-1}B_1w(s) + (sI-A)^{-1}B_2K(s)x(s), \\ \big(I-(sI-A)^{-1}B_2K(s)\big)\, x(s) &= (sI-A)^{-1}B_1w(s). \end{aligned}$$

In order for the system to be well-posed $(I-(sI-A)^{-1}B_2K(s))$ must be invertible. Since $K(s)$ is proper, see Definition 2.4.3, $(I-(sI-A)^{-1}B_2K(s))$ is invertible for large enough s. Hence

$$\begin{aligned} x(s) &= (I-(sI-A)^{-1}B_2K(s))^{-1}(sI-A)^{-1}B_1w(s) && (3.83)\\ &= ((sI-A)^{-1}(sI-A)-(sI-A)^{-1}B_2K(s))^{-1}(sI-A)^{-1}B_1w(s) && (3.84)\\ &= ((sI-A)^{-1}(sI-A-B_2K(s)))^{-1}(sI-A)^{-1}B_1w(s) && (3.85)\\ &= (sI-A-B_2K(s))^{-1}(sI-A)(sI-A)^{-1}B_1w(s) && (3.86)\\ &= (sI-A-B_2K(s))^{-1}B_1w(s). && (3.87) \end{aligned}$$

Therefore $$T_{wx} = (sI-A-B_2K(s))^{-1}B_1.$$ □

Lemma 3.2.8. Let G, K, u and X be defined as in equations (3.4), (2.34) and (3.2.1) respectively. Suppose K is proper and stable, as in Definitions 2.4.3 and 2.4.5. Assume K stabilizes G. Define $v(s) = u(s) + B_2^*Xx(s)$. Let T_{wv} be the transfer function from w to v. Then T_{wv} is stable and strictly proper (see Definition 2.4.4).

Proof. Using Lemma 3.2.7 and (2.34)

$$\begin{aligned} v(s) &= K(s)x(s) + B_2^*Xx(s) && (3.88)\\ v(s) &= (K(s)+B_2^*X)T_{wx}w(s) && (3.89)\\ v(s) &= (K(s)+B_2^*X)(sI-A-B_2K(s))^{-1}B_1w(s). && (3.90) \end{aligned}$$

Therefore $T_{wv} = (K(s)+B_2^*X)(sI-A-B_2K(s))^{-1}B_1$. By Lemma 3.2.7, $(sI-A-B_2K(s))^{-1}B_1$ is T_{wx}. Because $K(s)$ stabilizes G, $(sI-A-B_2K(s))^{-1}B_1$ is stable, as in Definition 2.4.5. Because $K(s)$ and $(sI-A-B_2K(s))^{-1}B_1$ are stable, T_{wv} is stable.

To show that T_{wv} is strictly proper, as in Definition 2.4.4, it is required to establish that

$$\lim_{|s|\to\infty}(K(s)+B_2^*X)(sI-A-B_2K(s))^{-1} = 0. \qquad (3.91)$$

Since $K(s)$ is proper, there exists a constant matrix D such that $\lim_{|s|\to\infty} K(s) = D$. For large enough $|s|$, $(sI - A - B_2K(s))$ is invertible. For invertible matrices, matrix inversion is continuous. Therefore

$$\lim_{|s|\to\infty} (K(s) + B_2^*X)\,(sI - A - B_2K(s))^{-1} \tag{3.92}$$

$$= \left(\lim_{|s|\to\infty} K(s) + B_2^*X\right)\left(\lim_{|s|\to\infty} \frac{1}{s}\right)\left(I - \lim_{|s|\to\infty} \frac{A + B_2K(s)}{s}\right)^{-1} \tag{3.93}$$

$$= (D + B_2^*X)\left(\lim_{|s|\to\infty} \frac{1}{s}\right) I \tag{3.94}$$

$$= 0. \qquad \square$$

The following theorem gives the optimal controller and the optimal cost for the state feedback problem.

Theorem 3.2.9. (State Feedback Problem) Let the generalized plant G have the structure (3.4) and let $X, A_F, C_F, U, U^\sim$ and G_c be defined by Equations (3.7), (3.8), (3.9), (3.32) and (3.33). Then
1. the H_2-optimal controller as defined in (3.3) is the constant gain $K = -B_2^*X$,
2. the H_2-optimal cost as defined in (3.2) is $\sqrt{\text{tr}(B_1^*XB_1)}$.

Proof. Let $u(t)$ be the output of any stabilizing controller. Define $v(t) = u(t) + (B_2^*Xx(t))$. Then the state space equations can be written as

$$\begin{aligned}\dot{x} &= Ax + B_1w + B_2(v - B_2^*Xx) \\ &= (A - B_2B_2^*X)x + B_1w + B_2v \\ &= A_Fx + B_1w + B_2v.\end{aligned} \qquad \begin{aligned}z &= C_1x + D_{12}(v - B_2^*Xx) \\ &= (C_1 - D_{12}B_2^*X)x + D_{12}v \\ &= C_Fx + D_{12}v.\end{aligned}$$

Using (3.33), z is written in terms of w and v in the frequency domain.

$$z = C_F(sI - A_F)^{-1}B_1w + (C_F(sI - A_F)^{-1}B_2 + D_{12})v, \tag{3.95}$$

$$= G_cB_1w + Uv. \tag{3.96}$$

Let T_{vw} be the transfer function from w to v. By Lemma, 3.2.8 T_{vw} is strictly proper and stable. Therefore it is in RH_2 (Definition 2.4.1). Based on (3.96), z can be written as

$$z = (G_cB_1 + UT_{vw})w. \tag{3.97}$$

Therefore the closed-loop transfer function is $T_{zw} = G_cB_1 + UT_{vw}$. Using Definition 2.5.1, the H_2-norm of T_{zw} is calculated as an integral over the imaginary axis using Definition (2.103).

$$\begin{aligned} ||T_{zw}||_{H_2}^2 &= \frac{1}{2\pi}\,\mathrm{tr}\left(\int_{-\infty}^{\infty}(G_cB_1 + UT_{vw})^*(G_cB_1 + UT_{vw})d\omega\right) \qquad (3.98)\\ &= \frac{1}{2\pi}\,\mathrm{tr}\left(\int_{-\infty}^{\infty}(B_1^*G_c^* + T_{vw}^*U^*)(G_cB_1 + UT_{vw})d\omega\right)\\ &= \frac{1}{2\pi}\,\mathrm{tr}(\int_{-\infty}^{\infty} B_1^*G_c^*G_cB_1 + B_1^*G_c^*UT_{vw} + T_{vw}^*U^*G_cB_1\\ &+ T_{vw}^*U^*UT_{vw}d\omega). \qquad (3.99)\end{aligned}$$

On the imaginary axis, $U^* = U^\sim$ because $s^* = -s$. Therefore $U^*G_c = U^\sim G_c$ on the imaginary axis. In (3.35) it was shown that $U^\sim G_c \in RH_2^\sim$. Since $T_{vw} \in RH_2$, Theorem 2.4.2 implies the integral of the terms $B_1^*G_c^*UT_{vw} + T_{vw}^*U^*G_cB_1$ is zero. Furthermore equation (3.34) implies $T_{vw}^*U^*UT_{vw} = T_{vw}^*T_{vw}$. The integral of the term $B_1^*G_c^*G_cB_1$ is equal to $||G_cB_1||_{H_2}^2$. Therefore

$$||T_{zw}||_{H_2}^2 = ||G_cB_1||_{H_2}^2 + ||T_{vw}||_{H_2}^2. \qquad (3.100)$$

The minimum of $||T_{zw}||_{H_2}^2$ is attained when $T_{vw} = 0$ and $v = 0$. If $v = 0$, then $u = -B_2^*Xx$. Therefore the optimal cost is $||G_cB_1||_{H_2}$ and the optimal controller is $K = -B_2^*X$. In Lemma 3.2.6, it was shown that $||G_cB_1||_{H_2}^2$ is equal to $\mathrm{tr}(B_1^*XB_1)$, therefore

$$||G_cB_1||_{H_2} = \sqrt{\mathrm{tr}(B_1^*XB_1)}. \qquad (3.101)$$

□

Note that the square of the H_2-optimal cost, $\mathrm{tr}(B_1^*XB_1)$, is similar to the optimal LQR cost $x_0^*Xx_0$, see [23, chapter 5, page 173]. The role of matrix B_1 is similar to x_0. In the generalized plant B_1 is related to the disturbance and in the LQR x_0 is the initial condition. The initial condition is always a column vector because it has the same size as the state variable. However the number of columns in B_1 corresponds to the number of the disturbances.

3.2.2 Full Information

The full information plant has the structure

$$G_{FI} = \left[\begin{array}{c|cc} A & B_1 & B_2 \\ \hline C_1 & 0 & D_{12} \\ \begin{bmatrix} I \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ I \end{bmatrix} & \begin{bmatrix} 0 \\ 0 \end{bmatrix} \end{array}\right]. \tag{3.102}$$

The measured output y is

$$\begin{aligned} y &= \begin{bmatrix} I \\ 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ I \end{bmatrix} w & (3.103) \\ &= \begin{bmatrix} x \\ w \end{bmatrix}. & (3.104) \end{aligned}$$

Note that both the state variable and the disturbance are available. This is the reason this problem is called full information.

Theorem 3.2.10. For the full information problem
1. the H_2-optimal controller is the constant gain $K = \begin{bmatrix} -B_2^* X & 0 \end{bmatrix}$.
2. the H_2-optimal cost is $\sqrt{\text{tr}(B_1^* X B_1)}$.

Proof. Similar to Theorem 3.2.9 define $v = u - (-B_2^* X x)$. By (3.97), the measured output z can be written as

$$z = (G_c B_1 + U T_{vw}) w.$$

By (3.100), the H_2-cost satisfies

$$||T_{zw}||_{H_2}^2 = ||G_c B_1||_{H_2}^2 + ||T_{vw}||_{H_2}^2. \tag{3.105}$$

Since the state variable x is available to the controller by (3.104), the controller output, u, can be set to $-B_2^* X x$ and $v = 0$. Therefore $T_{vw} = 0$ minimizes (3.105) the H_2-optimal cost is $||G_c B_1||_{H_2}^2$. By Lemma (3.2.6) the optimal cost is equal to $\sqrt{\text{tr}(B_1^* X B_1)}$. This proof is similar to the state feedback case because in both problems the state variable is available to the controller. However $K = \begin{bmatrix} -B_2^* X & 0 \end{bmatrix}$ has an additional zero compared to the state feedback optimal controller because K needs to be compatible with the dimension of y as required in the equation $u = Ky$. □

3.2.3 Full Control

The full control generalized plant has the following structure

$$G_{FC} = \left[\begin{array}{c|cc} A & B_1 & \left[\begin{array}{cc} I & 0 \end{array}\right] \\ \hline C_1 & 0 & \left[\begin{array}{cc} 0 & I \end{array}\right] \\ C_2 & D_{21} & 0 \end{array}\right]. \tag{3.106}$$

This structure is called full control because the control variable u can be separated into two parts $\left[\begin{array}{c} u_1 \\ u_2 \end{array}\right]$ such that u_1 only affects the dynamics and u_2 only affects the cost.

$$\begin{aligned} \dot{x} &= Ax + B_1 w + \left[\begin{array}{cc} I & 0 \end{array}\right] \left[\begin{array}{c} u_1 \\ u_2 \end{array}\right] & \qquad z &= C_1 x + \left[\begin{array}{cc} 0 & I \end{array}\right] \left[\begin{array}{c} u_1 \\ u_2 \end{array}\right] \\ &= Ax + B_1 w + u_1, & &= C_1 x + u_2. \end{aligned}$$

It is possible to solve the full control problem using the full information result. Define

$$G_{FC^*} = \left[\begin{array}{c|cc} A^* & C_1^* & C_2^* \\ \hline B_1^* & 0 & D_{21}^* \\ \left[\begin{array}{c} I \\ 0 \end{array}\right] & \left[\begin{array}{c} 0 \\ I \end{array}\right] & 0 \end{array}\right]. \tag{3.107}$$

Lemma 3.2.11. Let $G_{FC} = \left[\begin{array}{cc} G_{11} & G_{12} \\ G_{21} & G_{22} \end{array}\right]$ as defined in equation (2.58). Then

$$G_{FC^*} = \left[\begin{array}{cc} G_{11}^* & G_{21}^* \\ G_{12}^* & G_{22}^* \end{array}\right]. \tag{3.108}$$

Proof. Using equations (2.58), (3.106) and (3.107)

$$\begin{aligned} G_{11_{FC}}^* &= (C_1(sI - A)^{-1} B_1 + D_{11})^* && (3.109) \\ &= B_1^*(sI - A^*)^{-1} C_1^* + D_{11}^* && (3.110) \\ &= G_{11_{FC^*}}. && (3.111) \end{aligned}$$

The proof for G_{12}, G_{21}, G_{22} is identical. □

Theorem 3.2.12. (Duality) Let $G = \left[\begin{array}{cc} G_{11} & G_{12} \\ G_{21} & G_{22} \end{array}\right]$, K be a generalized plant and a controller. Let T_1, T_2 be the closed-loop transfer functions of G coupled with K and $G^* = \left[\begin{array}{cc} G_{11}^* & G_{21}^* \\ G_{12}^* & G_{22}^* \end{array}\right]$ coupled with K^* respectively. Then $T_1^* = T_2$.

Proof. Using Equation (2.38), $T_1 = G_{11} + G_{12}(I - KG_{22})^{-1}KG_{21}$.

$$\begin{aligned}
T_1^* &= (G_{11} + G_{12}(I - KG_{22})^{-1}KG_{21})^* && (3.112)\\
&= G_{11}^* + G_{21}^*K^*(I - KG_{22})^{-1^*}G_{12}^* && (3.113)\\
&= G_{11}^* + G_{21}^*K^*(I - G_{22}^*K^*)^{-1}G_{12}^* && (3.114)\\
&= G_{11}^* + G_{21}^*IK^*(I - G_{22}^*K^*)^{-1}G_{12}^* && (3.115)\\
&= G_{11}^* + G_{21}^*(I - K^*G_{22}^*)^{-1}(I - K^*G_{22}^*)K^*(I - G_{22}^*K^*)^{-1}G_{12}^* && (3.116)\\
&= G_{11}^* + G_{21}^*(I - K^*G_{22}^*)^{-1}(K^* - K^*G_{22}^*K^*)(I - G_{22}^*K^*)^{-1}G_{12}^* && (3.117)\\
&= G_{11}^* + G_{21}^*(I - K^*G_{22}^*)^{-1}K^*(I - G_{22}^*K^*)(I - G_{22}^*K^*)^{-1}G_{12}^* && (3.118)\\
&= G_{11}^* + G_{21}^*(I - K^*G_{22}^*)^{-1}K^*G_{12}^* && (3.119)\\
&= T_2. && \square
\end{aligned}$$

The next theorem gives the optimal controller and the optimal cost for the full control problem.

Theorem 3.2.13. (Full Control Problem) Let G_{FC} be defined as in (3.106). Let Y be the unique stabilizing solution to the Riccati equation (see Lemma 3.2.2 and assumptions B1 and B4 in section 3.1).

$$AY + YA^* - YC_2^*C_2Y + B_1B_1^* = 0. \qquad (3.120)$$

Then

1. the H_2-optimal controller is $K = \begin{bmatrix} -YC_2^* \\ 0 \end{bmatrix}$.
2. the H_2-optimal cost is $\sqrt{\text{tr}(C_1YC_1^*)}$.

Proof. Theorem 3.2.12 implies that the optimal controller for G_{FC} is the transpose of the optimal controller for G_{FC^*} as defined in (3.107). Since G_{FC^*} is a full information plant, the optimal cost is $\sqrt{\text{tr}(C_1YC_1^*)}$. The optimal controller is

$$K_{FC^*} = \begin{bmatrix} -(C_2^*)^*Y & 0 \end{bmatrix}. \qquad (3.121)$$

where Y is defined in (3.120). Note that equation (3.7) changes to (3.120) under the substitutions $A \to A^*, B_2 \to C_2^*, C_1 \to B_1^*$. The optimal controller for the full control problem is

$$\begin{aligned}
K_{FC} &= K_{FC^*}^* && (3.122)\\
&= \begin{bmatrix} -Y^*C_2^* \\ 0 \end{bmatrix}, && (3.123)\\
&= \begin{bmatrix} -YC_2^* \\ 0 \end{bmatrix}. && (3.124)
\end{aligned}$$

The symmetry of Y is used in the last equation, see Lemma 3.2.1. □

3.2.4 Output Estimation

The generalized plant in the output estimation problem has the structure

$$G_{OE} = \left[\begin{array}{c|cc} A & B_1 & B_2 \\ \hline C_1 & 0 & I \\ C_2 & D_{21} & 0 \end{array}\right]. \tag{3.125}$$

It is possible to reduce the output estimation problem to the full control problem by coupling its plant with an auxiliary system. An additional assumption that $A - B_2C_1$ is Hurwitz is needed. See Figure 3.3.

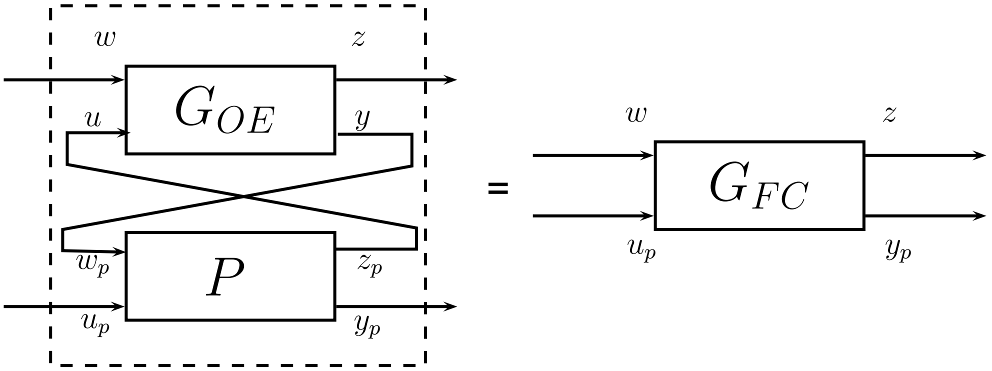


Figure 3.3: On the left hand side there is an Output Estimation (OE) generalized plant that is coupled to an auxiliary plant P. The combined system is equivalent to a Full Control (FC) plant.

By (3.125), the outputs of G_{OE} are

$$z = C_1x + u, \tag{3.126}$$
$$y = C_2x + D_{21}w. \tag{3.127}$$

To reduce the cost, u must be an estimate of $-C_1x$. Therefore this problem is called output estimation.

System P has the following structure. Note that P is stable if $A - B_2C_1$ is Hurwitz.

$$P = \left[\begin{array}{c|cc} A - B_2C_1 & 0 & \left[\begin{array}{cc} I & -B_2 \end{array}\right] \\ \hline C_1 & 0 & \left[\begin{array}{cc} 0 & I \end{array}\right] \\ C_2 & I & \left[\begin{array}{cc} 0 & 0 \end{array}\right] \end{array}\right]. \tag{3.128}$$

Theorem 3.2.14. Let G_{FC} and G_{OE} be defined as (3.106) and (3.125), then, the following relationships hold.

1. The plant G_{FC} can be transformed into G_{OE}.
2. G_{OE} coupled with P according to Figure 3.3 equals G_{FC}.

Proof. 1. The state space equations for G_{FC} and G_{OE} are

$$G_{OE} = \begin{cases} \dot{x} = Ax + B_1w + B_2u, \\ z = C_1x + u, \\ y = C_2x + D_{21}w. \end{cases} \qquad G_{FC} = \begin{cases} \dot{x} = Ax + B_1w + \left[\begin{array}{cc} I & 0 \end{array}\right]\left[\begin{array}{c} u_1 \\ u_2 \end{array}\right], \\ z = C_1x + \left[\begin{array}{cc} 0 & I \end{array}\right]\left[\begin{array}{c} u_1 \\ u_2 \end{array}\right], \\ y = C_2x + D_{21}w. \end{cases}$$

By coupling the input to the G_{FC} with a the constant matrix $\left[\begin{array}{c} B_2 \\ I \end{array}\right]$, the input becomes

$$\left[\begin{array}{c} B_2 \\ I \end{array}\right] u = \left[\begin{array}{c} B_2u \\ u \end{array}\right]. \tag{3.129}$$

Let

$$\left[\begin{array}{c} u_1 \\ u_2 \end{array}\right] = \left[\begin{array}{c} B_2u \\ u \end{array}\right]. \tag{3.130}$$

Then the equations for G_{FC} become

$$\begin{aligned} \dot{x} &= Ax + B_1w + \left[\begin{array}{cc} I & 0 \end{array}\right]\left[\begin{array}{c} u_1 \\ u_2 \end{array}\right] & \qquad z &= C_1x + \left[\begin{array}{cc} 0 & I \end{array}\right]\left[\begin{array}{c} u_1 \\ u_2 \end{array}\right] \\ \dot{x} &= Ax + B_1w + \left[\begin{array}{cc} I & 0 \end{array}\right]\left[\begin{array}{c} B_2u \\ u \end{array}\right] & \qquad z &= C_1x + \left[\begin{array}{cc} 0 & I \end{array}\right]\left[\begin{array}{c} B_2u \\ u \end{array}\right] \\ \dot{x} &= Ax + B_1w + B_2u, & \qquad z &= C_1x + u, \end{aligned}$$

which are the equations for G_{OE}.

2. Let x and p be the state variables of the systems G_{OE} and P respectively. Let the input to P be $u_p = \begin{bmatrix} u_{p1} \\ u_{p2} \end{bmatrix}$. Using (3.125) and (3.128), the state space equations for both systems are

$$G_{OE} = \begin{cases} \dot{x} = Ax + B_1 w + B_2 u \\ z = C_1 x + u \\ y = C_2 x + D_{21} w, \end{cases} \tag{3.131}$$

$$P = \begin{cases} \dot{p} = (A - B_2 C_1)p + \begin{bmatrix} I & -B_2 \end{bmatrix} \begin{bmatrix} u_{p1} \\ u_{p2} \end{bmatrix} \\ z_p = C_1 p + \begin{bmatrix} 0 & I \end{bmatrix} \begin{bmatrix} u_{p1} \\ u_{p2} \end{bmatrix} \\ y_p = C_2 p + w_p. \end{cases} \tag{3.132}$$

The two systems are connected as in Figure 3.3 using

$$u = z_p, \qquad\qquad w_p = y. \tag{3.133}$$

Substitute equation (3.133) into (3.131), (3.132) and simplify to arrive at

$$\begin{aligned} \dot{x} &= Ax + B_1 w + B_2 C_1 p + B_2 u_{p2}, \\ z &= C_1(x + p) + u_{p2}, \\ y_p &= C_2 p + C_2 x + D_{21} w. \end{aligned}$$

For a system $\widetilde{G}$ with the joint state variable $\begin{bmatrix} x \\ p \end{bmatrix}$, inputs w, u_p and outputs z and y_p the state space equations are

$$\begin{bmatrix} \dot{x} \\ \dot{p} \end{bmatrix} = \begin{bmatrix} A & B_2 C_1 \\ 0 & A - B_2 C_1 \end{bmatrix} \begin{bmatrix} x \\ p \end{bmatrix} + \begin{bmatrix} B_1 \\ 0 \end{bmatrix} w + \begin{bmatrix} 0 & B_2 \\ I & -B_2 \end{bmatrix} \begin{bmatrix} u_{p1} \\ u_{p2} \end{bmatrix} \tag{3.134}$$

$$z = \begin{bmatrix} C_1 & C_1 \end{bmatrix} \begin{bmatrix} x \\ p \end{bmatrix} + \begin{bmatrix} 0 & I \end{bmatrix} \begin{bmatrix} u_{p1} \\ u_{p2} \end{bmatrix} \tag{3.135}$$

$$y_p = \begin{bmatrix} C_2 & C_2 \end{bmatrix} \begin{bmatrix} x \\ p \end{bmatrix} + D_{21} w. \tag{3.136}$$

Defining $H = \begin{bmatrix} I & -I \\ 0 & I \end{bmatrix}$, apply the state transformation $H^{-1} = \begin{bmatrix} I & I \\ 0 & I \end{bmatrix}$ and note $H^{-1} \begin{bmatrix} x \\ p \end{bmatrix} = \begin{bmatrix} x+p \\ p \end{bmatrix}$. Using (3.47), equation (3.134), (3.135) and (3.136) are transformed as

$$\begin{aligned}
\begin{bmatrix} \dot{x}+\dot{p} \\ \dot{p} \end{bmatrix} &= H^{-1} \begin{bmatrix} A & B_2C_1 \\ 0 & A - B_2C_1 \end{bmatrix} H \begin{bmatrix} x+p \\ p \end{bmatrix} \\
&\quad + H^{-1} \begin{bmatrix} B_1 \\ 0 \end{bmatrix} w + H^{-1} \begin{bmatrix} 0 & B_2 \\ I & -B_2 \end{bmatrix} \begin{bmatrix} u_{p1} \\ u_{p2} \end{bmatrix}, && (3.137) \\
&= \begin{bmatrix} A & 0 \\ 0 & A - B_2C_1 \end{bmatrix} \begin{bmatrix} x+p \\ p \end{bmatrix} + \begin{bmatrix} B_1 \\ 0 \end{bmatrix} w + \begin{bmatrix} I & 0 \\ I & -B_2 \end{bmatrix} \begin{bmatrix} u_{p1} \\ u_{p2} \end{bmatrix}. && (3.138)
\end{aligned}$$

$$\begin{aligned}
z &= \begin{bmatrix} C_1 & C_1 \end{bmatrix} H^{-1} \begin{bmatrix} x+p \\ p \end{bmatrix} + \begin{bmatrix} 0 & I \end{bmatrix} \begin{bmatrix} u_{p1} \\ u_{p2} \end{bmatrix}, && (3.139) \\
&= \begin{bmatrix} C_1 & 0 \end{bmatrix} \begin{bmatrix} x+p \\ p \end{bmatrix} + \begin{bmatrix} 0 & I \end{bmatrix} \begin{bmatrix} u_{p1} \\ u_{p2} \end{bmatrix}. && (3.140)
\end{aligned}$$

$$\begin{aligned}
y_p &= \begin{bmatrix} C_2 & C_2 \end{bmatrix} H^{-1} \begin{bmatrix} x+p \\ p \end{bmatrix} + D_{21}w, && (3.141) \\
&= \begin{bmatrix} C_2 & 0 \end{bmatrix} \begin{bmatrix} x+p \\ p \end{bmatrix} + D_{21}w. && (3.142)
\end{aligned}$$

Therefore

$$\widetilde{G} = \left[\begin{array}{c|cc} \begin{bmatrix} A & 0 \\ 0 & A - B_2C_1 \end{bmatrix} & \begin{bmatrix} B_1 \\ 0 \end{bmatrix} & \begin{bmatrix} I & 0 \\ I & -B_2 \end{bmatrix} \\ \hline \begin{bmatrix} C_1 & 0 \end{bmatrix} & 0 & \begin{bmatrix} 0 & I \end{bmatrix} \\ \begin{bmatrix} C_2 & 0 \end{bmatrix} & D_{21} & 0 \end{array}\right]. \tag{3.143}$$

Since the second components of the observation matrices,

$$\begin{bmatrix} C_1 & 0 \\ C_2 & 0 \end{bmatrix}, \tag{3.144}$$

are zero, only $x+p$, not p, affects the output. Therefore, $\widetilde{G}$ can be written as G_{FC}, (3.106),

$$\widetilde{G} = \left[\begin{array}{c|cc} A & B_1 & \begin{bmatrix} I & 0 \end{bmatrix} \\ \hline C_1 & 0 & \begin{bmatrix} 0 & I \end{bmatrix} \\ C_2 & D_{21} & 0 \end{array}\right]. \tag{3.145}$$

The transfer functions corresponding to (3.143) and (3.145), calculated according to Definition 3.2.1, are equal. This is why (3.143) can be written as (3.145). □

The next theorem establishes the optimal controller and the optimal cost for the output feedback problem.

Theorem 3.2.15. Let G be a generalized plant with the output estimation structure (3.125). Let Y be the stabilizing solution to (3.120). Then

1. the optimal controller is $K(s) = \left[\begin{array}{c|c} A - B_2C_1 - YC_2^*C_2 & -YC_2^* \\ \hline C_1 & 0 \end{array}\right]$,
2. the optimal cost is $\sqrt{\operatorname{tr}(C_1YC_1^*)}$.

Proof. 1. Based on part 2 of Theorem 3.2.14, G_{OE} coupled with the plant P is equivalent to G_{FC}, see Figure 3.3. Denote the optimal controller for G_{FC} as K_{FC}. Coupling K_{FC} with G_{FC} is equivalent to coupling K_{FC} with P and then with G_{OE}, see Figure 3.4. Therefore the optimal controller for G_{OE}, denoted by K_{OE}, is K_{FC} coupled with P.

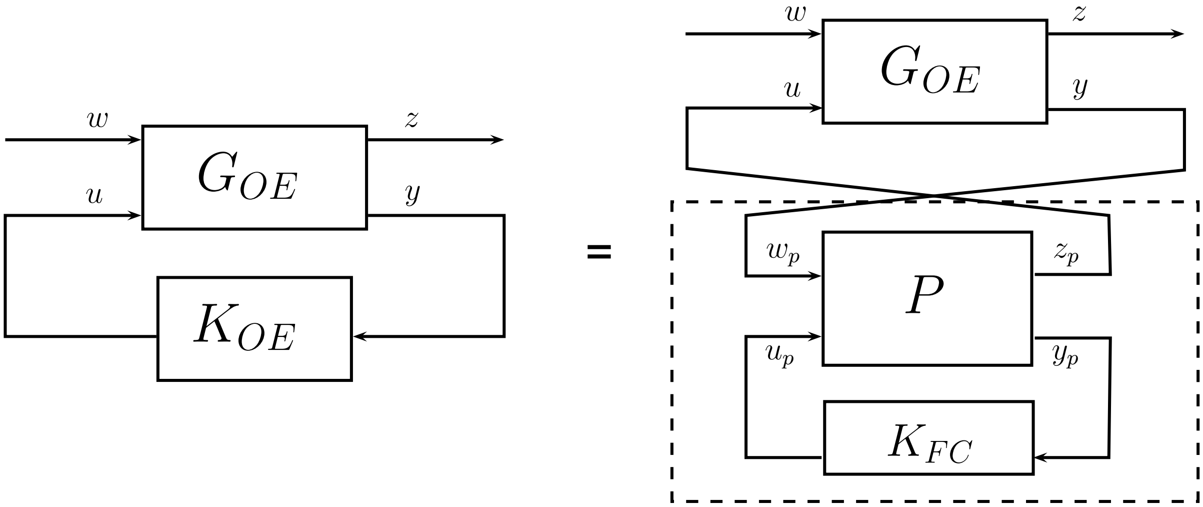


Figure 3.4: The optimal output estimation controller K_{OE} is the optimal full control controller coupled with the system P.

Using (3.124) and (3.128), the state space equations for P are

$$\begin{aligned}
\dot{p} &= (A - B_2C_1)p + \left[\begin{array}{cc} I & -B_2 \end{array}\right] u_p, \\
z_p &= C_1p + \left[\begin{array}{cc} 0 & I \end{array}\right] u_p, \\
y_p &= C_2p + w_p, && (3.146) \\
& && (3.147)
\end{aligned}$$

and

$$K_{FC} = \left[\begin{array}{c} -YC_2^* \\ 0 \end{array}\right]. \tag{3.148}$$

From Figure 3.4,

$$\begin{aligned}
u_p &= K_{FC}\ \ y_p, && (3.149) \\
&= \left[\begin{array}{c} -YC_2^* \\ 0 \end{array}\right] (C_2p + w_p), && (3.150) \\
&= \left[\begin{array}{c} -YC_2^*C_2p - YC_2^*w_p \\ 0 \end{array}\right]. && (3.151)
\end{aligned}$$

Substitute (3.151) into (3.146) and note that $w_p = y$ and $u = z_p$,

$$\begin{aligned}
\dot{p} &= (A - B_2C_1)p + \left[\begin{array}{cc} I & -B_2 \end{array}\right] \left[\begin{array}{c} -YC_2^*C_2p - YC_2^*y \\ 0 \end{array}\right] && (3.152) \\
&= (A - B_2C_1)p - YC_2^*C_2p - YC_2^*y && (3.153) \\
&= (A - B_2C_1 - YC_2^*C_2)p - YC_2^*y. && (3.154)
\end{aligned}$$

Also

$$\begin{aligned}
u &= z_p && (3.155) \\
&= C_1p + \left[\begin{array}{cc} 0 & I \end{array}\right] u_p && (3.156) \\
&= C_1p + \left[\begin{array}{cc} 0 & I \end{array}\right] \left[\begin{array}{c} -YC_2^*C_2p - YC_2^*w_p \\ 0 \end{array}\right] && (3.157) \\
&= C_1p. && (3.158)
\end{aligned}$$

Therefore, using (3.154) and (3.158), K_{OE} with the input y, output u and state p is

$$K_{OE}(s) = \left[\begin{array}{c|c} A - B_2C_1 - YC_2^*C_2 & -YC_2^* \\ \hline C_1 & 0 \end{array}\right]. \tag{3.159}$$

Since coupling of K_{OE}, G_{OE} is equivalent to coupling of K_{FC}, G_{FC} the the optimal cost for the output estimation problem is the same as that for the full control problem. Using part 2 of Theorem 3.2.13, the optimal cost is $\sqrt{\mathrm{tr}(C_1YC_1^*)}$. □

3.2.5 Output Feedback

The output feedback problem is the most general case.

$$G_{OF} = \left[\begin{array}{c|cc} A & B_1 & B_2 \\ \hline C_1 & 0 & D_{12} \\ C_2 & D_{21} & 0 \end{array}\right]. \tag{3.160}$$

The following theorem gives the optimal controller and the optimal cost for the output feedback problem. For the ease of reference and convenience the Riccati equations (3.120) and (3.120) are repeated here.

Theorem 3.2.16. Let the generalized plant G have the structure (3.160). Let X and Y, respectively, be the stabilizing solutions of the following Riccati equations.

$$\begin{aligned} A^*X + XA - XB_2B_2^*X + C_1^*C_1 &= 0, \\ AY + YA^* - YC_2^*C_2Y + B_1B_1^* &= 0. \end{aligned}$$

Then

1. the optimal controller is $K = \left[\begin{array}{c|c} A - B_2B_2^*X - YC_2^*C_2 & YC_2^* \\ \hline -B_2^*X & 0 \end{array}\right]$,
2. the optimal cost is $\sqrt{\mathrm{tr}(B_1^*XB_1) + \mathrm{tr}(B_2^*XYXB_2)}$.

Proof. Similar to the argument in Theorem 3.2.9, define $v = u + (B_2^*X)x$ and rewrite the state space equations of G_{OF} in terms of v.

$$\begin{aligned} \dot{x} &= Ax + B_1w + B_2(v - B_2^*Xx), & z &= C_1x + D_{12}(v - B_2^*Xx), \\ &= (A - B_2B_2^*X)x + B_1w + B_2v, & &= (C_1 - D_{12}B_2^*X)x + D_{12}v, \\ &= A_Fx + B_1w + B_2v. & &= C_Fx + D_{12}v. \end{aligned}$$

Using equation (3.33), z is written in the frequency domain as

$$z = C_F(sI - A_F)^{-1}B_1w + (C_F(sI - A_F)^{-1}B_2 + D_{12})v, \tag{3.161}$$

$$= G_cB_1w + Uv, \tag{3.162}$$

where G_c and U are defined in (3.33). Using (3.100) and (3.101),

$$||T_{zw}||_{H_2}^2 = ||G_cB_1||_{H_2}^2 + ||T_{vw}||_{H_2}^2, \tag{3.163}$$

$$= \mathrm{tr}(B_1^*XB_1) + ||T_{vw}||_{H_2}^2. \tag{3.164}$$

Because of (3.164) minimizing $||T_{zw}||_{H_2}^2$ is equivalent to minimizing $||T_{vw}||_{H_2}^2$. In the state feedback problem it was possible to put $T_{vw} = 0$ because choosing $u = -B_2^* X x$ results in $v = 0$. However in the output feedback problem this is not possible because the state variable is not available for feedback. Minimizing $||T_{vw}||_{H_2}^2$ is an output estimation problem because a plant can be created with v as the output. Consider the equations

$$\begin{aligned} \dot{x} &= Ax + B_1 w + B_2 u, \\ v &= B_2^* X x + u, \\ y &= C_2 x + D_{21} w, && (3.165) \\ & && (3.166) \end{aligned}$$

and define

$$G_v = \left[\begin{array}{c|cc} A & B_1 & B_2 \\ \hline B_2^* X & 0 & I \\ C_2 & D_{21} & 0 \end{array}\right] \qquad (3.167)$$

which has the output estimation structure, see (3.125). The output estimation problem requires the stability of $A - B_2(B_2^* X)$. This is guaranteed by Lemma 3.2.1. Based on Theorem 3.2.15, the optimal cost for G_v is

$$||T_{vw}||_{H_2}^2 = \mathrm{tr}(B_2^* X Y X^* B_2). \qquad (3.168)$$

Returning to equation (3.164) and using (3.168)

$$\begin{aligned} ||T_{zw}||_{H_2}^2 &= ||G_c B_1||_{H_2}^2 + ||T_{vw}||_{H_2}^2 && (3.169) \\ &= \mathrm{tr}(B_1^* X B_1) + \mathrm{tr}(B_2^* X Y X^* B_2). && (3.170) \end{aligned}$$

Since $X = X^*$ by Lemma 3.2.1, the X^* in (3.170) can be replaced by X. The square root of (3.170) is the optimal output feedback cost.

The H_2-optimal controller for G_{OF} must satisfy equation (3.168) otherwise it can not achieve the minimum cost in equation (3.170). This controller is the optimal controller for G_v (Theorem 3.2.15) with the substitution $C_1 \to B_2^* X$,

$$K(s) = \left[\begin{array}{c|c} A - B_2 B_2^* X - Y C_2^* C_2 & Y C_2^* \\ \hline -B_2^* X & 0 \end{array}\right]. \qquad (3.171)$$

□

The optimal controller $K(s)$ in equation 3.171 has a special structure. Let $\hat{x}$ be the state variable of K, then the state space equations for K can be written as

$$\begin{aligned}\dot{\hat{x}} &= (A - B_2B_2^*X - YC_2^*C_2)\hat{x} + YC_2^*y, & (3.172)\\ &= A\hat{x} + B_2u + YC_2^*(y - C_2\hat{x}), & (3.173)\end{aligned}$$

$$u = -B_2^*X\hat{x}. \tag{3.174}$$

Equation (3.173) is an observer, see Theorem 2.1.2, and $\hat{x}$ is an estimate of the state x. The output of the controller, u, is the optimal feedback gain from Theorem 3.2.9 multiplied by the state estimate, $\hat{x}$. The state estimation and the construction of u are done separately from each other. This property is called separation. A controller with the separation property, first optimally estimates the state and then applies optimal state feedback.

Chapter 4

H_2-Optimal Sensor Location

In this chapter the optimal sensor location problem is defined formally. Using the H_2-cost function discussed in the previous chapter, different sensor locations can be compared to each other where the position corresponding to the lowest cost is the optimal. A sensor is confined to a region Ω of dimension less than or equal to three. If there are m sensors then the space of possibilities is Ω^m. If Ω is neither closed nor bounded, then the optimal location may not necessarily be in Ω. Therefore, in order to guarantee the existence of an optimal location, Ω needs to be closed and bounded. From a practical point of view, the boundedness of Ω is guaranteed, but there can be cases where Ω is open. For example, consider a sensor that can be placed anywhere on a beam except at the very end due to safety reasons. However the length of the beam minus an epsilon can be considered as Ω.

4.1 Problem Statement

Problem. Let $r \in \Omega^m \subset R^{3m}$ where Ω is a compact set. Let

$$C_2(r) : \Omega \to R^{m \times n} \tag{4.1}$$

be a continuous function and

$$G(s,r) = \left[\begin{array}{c|cc} A & B_1 & B_2 \\ \hline C_1 & 0 & D_{12} \\ C_2(r) & D_{21} & 0 \end{array} \right] \tag{4.2}$$

be a generalized plant that satisfies the assumptions in section 3.1. Let $X, Y(r)$ be the stabilizing, positive semi-definite solutions to the Riccati equations

$$A^*X + XA - XB_2B_2^*X + C_1^*C_1 = 0, \tag{4.3}$$
$$AY + YA^* - YC_2^*(r)C_2(r)Y + B_1B_1^* = 0. \tag{4.4}$$

Define the functions $J_1, J_2 : \Omega \to \mathbb{R}^+$ as the square of the full control and output feedback costs respectively (Theorems 3.2.13 and 3.2.16):

$$J_1(r) = \operatorname{tr}\left(C_1Y(r)C_1^*\right), \tag{4.5}$$
$$J_2(r) = \operatorname{tr}\left(B_1^*XB_1\right) + \operatorname{tr}\left(B_2^*XY(r)XB_2\right). \tag{4.6}$$

Define

$$r_1^* = \arg \inf_{r\in\Omega^m} J_1(r), \tag{4.7}$$
$$r_2^* = \arg \inf_{r\in\Omega^m} J_2(r). \tag{4.8}$$

The problem is to find r_1^* and r_2^*.

There could be multiple locations that correspond to the infimum; each is a valid solution. Continuity of J_1 and J_2 and compactness of Ω guarantee the existence of r_1^* and r_2^*. If the focus is only on state estimation, J_1 should be utilized as J_1 only considers the sensor (the actuator or X are not involved). If the overall performance of the actuator and the state estimation requires consideration, then J_2 should be used. The first term in J_2, $\operatorname{tr}(B_1^TXB_1)$, does not depend on r and can be removed. The second term in J_2 includes X, Y and B_2. Since B_2 and X depend on the actuator, the location of the actuator affects the optimal sensor location.

4.2 Literature Review

In this section a number of papers that have investigated the optimal sensor location are reviewed. Some papers describing the optimal actuator location have also been included because the two problems have many similarities and often involve similar techniques. The discussion is separated into three cases. The first subsection considers the closed-loop H_2-cost function where the H_2-optimal control is fixed as the control strategy and the optimal sensor location is found afterwards. In the second subsection, the open-loop H_2-costs are discussed which do not consider any particular control strategy and focus on the natural dynamics. Finally, the third subsection investigates non-H_2 methods such as the Linear- Quadratic (LQ) cost and costs based on the observability Gramian.

In a broader context, one must consider the final goal for optimizing the sensor location. According to Kubrusly [20] the optimal sensor or actuator location literature for systems governed by partial differential equations (PDEs) can be categorized into three major groups. The first group is system identification where sensors aid in identifying uncertain or unknown parameters in different models. The second group is state estimation where in the presence of noise and disturbance, an optimally placed sensor reduces the estimation error to the optimal value. Lastly, the third group is the closed-loop cost function where a control law such as LQR, H_2 or H_∞ is used. The control law has an associated cost function which reflects the performance of the controlled system with sensors and actuators optimized with respect to this cost.

4.2.1 Closed-Loop H_2-cost

In many practical problems the governing equations are PDEs where the independent variables are continuous in both time and space. Such systems will inherently have an infinite-dimensional state variable. Typically, the space variable is discretized so that problem is reduced to a finite dimension. If more accuracy is needed, the dimension of the approximation is increased. However, without convergence results from the sequence of the approximated problems can change drastically when higher dimensions are considered [24, Example 3.2, Figure 8]. Uniform exponential stabilizability and uniform exponential detectability of the sequence of matrices (A_n, B_n, C_n) for the finite-dimensional system of size n as $n \to \infty$ are required conditions for convergence of the results of the approximated problems.

In [22], the H_2-optimal theory is discussed for infinite-dimensional systems. Additional development can be found in [24]. The results are analogous to the finite-dimensional case with additional regularity conditions on the operators in the H_2-cost. The discussion focuses on the properties of the the semigroup generated by the operator A, whether or not operators B_1 and C_1 are compact, convergence of finite-dimensional approximations to the infinite-dimensional Riccati operator, convergence of the optimal locations, convergence of the optimal costs and continuity of the norm of the Riccati operator with respect to the location of the actuator. In the results section, two systems are considered: the one-dimensional damped beam and two-dimensional diffusion. The beam is cantilevered; such a beam has one fixed end and one free end. Two possibilities are considered for the state cost; either all states are weighted equally or only the deflection of the beam at the free end is considered. Regarding the disturbance, six different shapes of spatial distribution are investigated.

Results illustrate that the optimal actuator location often matches the maximum value of the disturbance shape, but not always. The worst case disturbance is also considered and the optimal actuator location is optimized over all possible spatial distributions of the disturbance. The eigenfunction of the Riccati operator with the largest value is considered to be the worst spatial distribution. For the two-dimensional diffusion, the effect of a variable diffusivity function is investigated. Two types of state weights are considered. The first weight is the identity for all states. The second weight averages over all of the states. The results suggest that the optimal actuator location is in the disturbance region if the disturbance is in a low diffusivity region. A similar discussion is present in [25] for the LQR problem and is discussed in section 4.2.3.

In [6], an application of H_2-optimal actuator and sensor location in fluid mechanics is considered. The governing equation is the linearized Ginzburg-Landau equation that describes velocity perturbations in a flow with the domain as the real line extended to infinity at both ends. Stochastic flow disturbances are also introduced in the spatial domain and in the measurement of the sensor.

The Ginzburg-Landau (GL) equation involves a complex variable q where the real part of q describes the velocity perturbation amplitudes:

$$\frac{\partial q}{\partial t} + \nu \frac{\partial q}{\partial x} = \mu(x) q + \gamma \frac{\partial^2 q}{\partial x^2} - a|q|^2 q, \tag{4.9}$$

with the boundary conditions $q = 0$ at both ends of the line and the flow from left to right. The non-linear term is linearized around the equilibrium $q = 0$ which leads to $|q|^2 q \approx 0$. The resulting equation can be rewritten as

$$\dot{q} = Lq, \tag{4.10}$$

$$L = -\nu \frac{\partial}{\partial x} + \mu(x) + \gamma \frac{\partial^2}{\partial x^2}. \tag{4.11}$$

With a parabolic form assumed for $\mu(x)$ the eigenfunctions of the operator L are

$$\phi_n(x) = e^{\frac{1}{2}(\frac{\nu x}{\gamma} - \chi^2 x^2)} H_n(\chi x). \tag{4.12}$$

where H_n is the nth Hermite polynomial and χ is a scaling factor which depends on the model parameters. The domain is discretized using $N = 100, q = \begin{bmatrix} q_1 & \cdots & q_N \end{bmatrix}^T$

grid points which are the roots of $H_N(\chi x)$(spanning [-56.06, 56.06]). A discrete spectral form is found for $\frac{\partial}{\partial x}, \frac{\partial^2}{\partial x^2}$ and L. Equations (4.10) is rewritten as

$$\dot{q} = Aq. \tag{4.13}$$

where A is the linearized discrete GL operator.

In order to create a generalized plant actuators, sensors, disturbance, noise and cost weights are required. Each sensor or actuator has a Gaussian shape with variance $\sigma = 0.4$. Let B_2, C_2 be column vectors that represent a Gaussian function centered at x_a and x_c respectively. Let d and n be column vectors of independent white Gaussian noise processes. Let $\beta = 7$ be a parameter that relates the relative weight of q in the cost function compared to the control variable u. The value of β is a design voice. A generalized plant is created using the following definitions. Define M as a diagonal matrix corresponding to numerical integration using the trapezoid rule. Additionally, let M have the square root $M^{1/2}$ with the square roots of the elements of M on the diagonal.

$$B_1 = \begin{bmatrix} I & 0 \end{bmatrix}, \quad C_1 = \begin{bmatrix} \beta M^{1/2} \\ 0 \end{bmatrix}, \quad D_{12} = \begin{bmatrix} 0 \\ I \end{bmatrix}, \quad D_{21} = \begin{bmatrix} 0 & 2 \times 10^{-4} I \end{bmatrix}, \tag{4.14}$$

$$z = \begin{bmatrix} \beta M^{1/2} q \\ u \end{bmatrix}, \quad w = \begin{bmatrix} d \\ n \end{bmatrix}, \tag{4.15}$$

$$\begin{bmatrix} \dot{q} \\ z \\ y \end{bmatrix} = \left[\begin{array}{c|cc} A & B_1 & B_2 \\ \hline C_1 & 0 & D_{12} \\ C_2 & D_{21} & 0 \end{array}\right] \begin{bmatrix} q \\ w \\ u \end{bmatrix}. \tag{4.16}$$

The right element of D_{21} is set to 2×10^{-4} to account for the minimal sensor noise. The control u is given by the H_2-optimal controller, see Theorem 3.2.16. The cost is the H_2-norm of the closed-loop system as a function of the actuator and the sensor location and is denoted by $\Gamma(x_a, x_s)$.

A gradient based approach is used to find the minimum of Γ. Algebraic equations necessary for any local minimum and an expression for the derivative of Γ with respect to x_a and x_s are derived. Using the conjugate gradient method, the values of (x_a, x_s) are iteratively updated until a local minimum is reached. This process is repeated with different initial conditions until results become consistent such that the final result can be accepted as the global minimum. However differentiability of Γ is not established. Furthermore, if the optimal location is on the boundary of the feasible region, the derivative of Γ is not necessarily zero, but this issue is also not pursued.

A pattern is observed where in the optimal configuration, for an equal number of actuators and sensors, each actuator is paired with a sensor such that the sensor is placed just downstream of the actuator.

Another example that uses the closed-loop H_2-cost is [10] however, this work utilizes a combination of the H_∞ and H_2 methods. The problem involves an undamped beam with n degrees of freedom and simply supported boundary conditions with two pairs of actuators and sensors. In comparison to [6], the sensors and actuators are collocated which means they have to be at the same location. An H_∞ controller is constructed that guarantees an H_∞ performance bound. The H_2-norm of the closed-loop system is then optimized with respect to the collocated actuator-sensor pairs. Similar to [6], algebraic equations are derived for the gradient of the H_2-norm, but no guarantees with respect to differentiability are offered. Unlike [6], a quasi-Newton method is used to iterate toward the optimal location. Since convergence can occur at local minima, the algorithm is repeated from different initial conditions until satisfactory results are obtained, however there is no discussion regarding what is considered satisfactory. A reasonable stopping condition is the consistency of results for a wide range of initial conditions. After the optimal locations are found, a performance comparison is done. The impulse responses of the uncontrolled beam, the controlled beam with sub-optimally placed and optimally placed actuator-sensor pairs are compared. The optimal placement performs more efficiently in terms of maximum deflection, settling time and other damping properties.

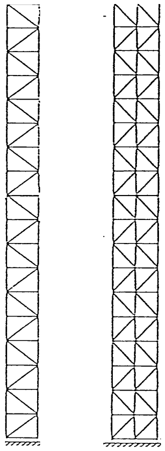


Figure 4.1: Mini-Mast truss from [2].

The final paper in the discussion of closed-loop H_2-cost is [2]. The paper discusses a twenty meter long truss structure cantilevered at its base, the NASA Langley Mini-Mast, see Figure 4.1. The structure was designed to represent future deployable space trusses. The truss has eighteen levels, also known as bays. There are total of sixty sensors on the structure but only a subset of ten are considered. There are four accelerometers and three displacement sensors at bay eighteen and three displacement sensors at bay ten. There are three shakers at bay nine that introduce vibrations. Additionally, there are three torque wheel actuators at the top, in bay eighteen. A finite element model of the truss is created with twenty eight modes, three control inputs, three disturbance inputs and ten outputs. A reduced-order model is also created with five modes.

Three performance objectives are considered: vibration reduction at bay ten, at bay eighteen, or at both bays. Five different combinations of sensors are compared with each other. The five groups are respectively, only accelerometers at bay eighteen, only displacement sensors at bay eighteen, both accelerometers and sensors at bay eighteen, displacement sensors at bay eighteen and ten and finally, all ten sensors. The results illustrate that accelerometers nearly provide full information performance. The displacement sensors at both bay ten and eighteen provide good performance however, placing displacement sensors at bay ten performs poorly in comparison to the other combinations. If the displacement sensors are collocated with the location where the performance is measured, then there is no major difference between accelerometers and displacement sensors. In the non-collocated case accelerometers outperform the displacement sensors. The results for the five mode model are similar to the model with the full order.

4.2.2 Open-Loop H_2-cost

In comparison to the closed-loop H_2-cost, there are alternative open-loop methods which do not consider a specific control strategy and only utilize the open-loop transfer functions.

In [1] the modal-H_2 approach is used for optimal sensor and actuator placement for the control of vibrations in a square carbon plate. The name modal comes from eigenfunction decomposition of the system, where each eigenfunction is also called a mode. Starting with the standard equation

$$\begin{aligned} \dot{x} &= Ax + w_1 + Bu, \\ y &= Cx + w_2, \end{aligned} \tag{4.17}$$

the authors exclude w_1 and w_2 in their analysis as w_1 and w_2 are exogenous and are generally not known ahead of time. Additionally, there is no discussion about the special case where w_1 and w_2 are white Gaussian. Let the size of x be n and let $m < n$ represent the first m modes that are of interest. A reduced model is considered where matrices A_{mi}, B_{mi}, C_{mi} are the reduced form for the ith coordinate given the first m modal coordinates. Assume that there are s different candidate positions for the sensor. Let $C_{(i,S)} = [C_{i1}, C_{i2}, ..., C_{is}]$ be the matrix describing the sensor when placed at each respective position. For each sensor position k, a modal

H_2-norm is defined as

$$||G_{ik}||_2 = \frac{||B_{mi}||_2||C_{mik}||_2}{2\sqrt{\zeta_i\omega_i}}, \tag{4.18}$$

where ζ_i is the damping ratio and ω_i is the natural frequency. Damping ratio and natural frequency are two fundamental properties of the second order systems. The interpretation of G_{ik} is the input-output norm of a controllable structure with one sensor at the kth place. To normalize the above norm, the sensor index is normalized as

$$\begin{gathered}\delta_{2,ik} = \frac{||G_{ik}||_2^2}{\max\limits_k ||G_{ik}||_2^2}, \\ i = 1, ..., m, \quad k = 1, ..., s.\end{gathered} \tag{4.19}$$

The index for the kth sensor is given by the weighted linear combination:

$$\delta(k) = \sum_{i=1}^{m} w_i\delta_{2,ik}, \tag{4.20}$$

where w_i are weights associated with different modes. The weights can be used to emphasize selected modes. For example, assigning one to the weight of the first five modes and zero for the other modes means that only the first five are considered. To reduce the the effect on a particular mode a weight of minus one can be assigned.

For the carbon plate model, there are five sensors which measure acceleration and five actuators which are piezoelectric patches. A genetic algorithm is used to guide the search. The genetic algorithm can stop at a local minimum. Furthermore the local minimum that is found depends on the initial condition of the algorithm. Therefore, the algorithm is reset multiple times until consistent results are obtained. The authors do not elaborate on how many resets were required and no conditions are mentioned under which the algorithm is guaranteed to produce consistent results, however, once consistent results are obtained they are assumed to be optimal. Two scenarios are considered. In scenario one, the goal is to control the first five modes therefore, only those five are assigned a positive weight and all others get a weight of zero. In the second scenario, the goal is to not only control the first five modes but also not to disturb the next five. The first five are assigned positive weights, where as the following five are assigned negative weights. For each scenario, the optimal sensor and actuator locations are found and validated utilizing different control strategies such as IMSC (Independent Modal Space Control). This approach is considered open-loop because the optimal locations are found before a control strategy is chosen. The authors note that IMSC usually has issues with higher uncontrolled modes, but it shows improvement when the actuators and sensors are placed according to the results in the second scenario.

The next method uses the so-called "spatial H_2-norm". In [9] this method is used to find the optimal actuator location for a diffusion problem in one dimension with Neumann boundary conditions. For infinite-dimensional systems that have continuous state variables, the transfer function from the input to the state variable is a function of both s and another variable that parametrizes the state, ζ. Let Ω be the domain for ζ. Let ζ_a be the actuator location, ϕ_i and λ_i be the system's eigenfunctions and eigenvalues respectively. Using eigenfunction expansion, a transfer function is defined as

$$G(s,\zeta,\zeta_a) = \sum_{i=1}^{\infty} \frac{B(\zeta_a)\phi(\zeta)}{s-\lambda_i}. \tag{4.21}$$

Let $f_i(\zeta_a)$ be the H_2-norm of $\dfrac{B(\zeta_a)\phi_i(\zeta)}{s-\lambda_i}$. Then two types of controllability indexes are defined. The ith modal controllability is

$$M_i(\zeta_a) = \frac{f_i(\zeta_a)}{\max\limits_{\zeta_a\in\Omega} f_i(\zeta_a)} \tag{4.22}$$

and the controllability index for the first N modes is defined as

$$S^N(\zeta_a) = \frac{\sqrt{\sum_{i=1}^{N} f_i^2(\zeta_a)}}{\max_{\zeta_a\in\Omega}\sqrt{\sum_{i=1}^{N} f_i^2(\zeta_a)}}. \tag{4.23}$$

The controllability index for the modes N to M is denoted by S_N^M. The authors pose the optimal actuator problem via maximizing the controllability index S^N while maintaining some design constrains on the individual modes and the controllability index of modes N to M. In other words

$$\begin{aligned} &\max_{\zeta_s\in\Omega} \quad S^N(\zeta_s), \\ &\text{subject to } M_i(\zeta_s) \geq \beta_i, \quad S_M^N(\zeta_s) \leq \gamma, \end{aligned} \tag{4.24}$$

where β_i and γ are chosen by the designer based on performance requirements. According to the numerical results, the optimal actuator location based on this approach tends to be close to the edge of the domain. As more modes are included, the optimal location moves closer towards the boundary.

4.2.3 Non-H_2 Cost

The Linear-Quadratic (LQ) cost function is another method for defining the optimal actuator and sensor location. In [8], [19] and [25] the LQ-cost has been used.

In [25], the theory for the existence of the optimal actuator location and the convergence of optimal location of the finite-dimensional approximations to the optimal location of the infinite-dimensional system are developed. The LQ-optimal cost depends on the initial condition and there are two approaches to deal with this dependence. One approach is to adopt the worst initial condition which is the one that corresponds the maximum cost. In this approach the norm of the Riccati operator becomes the optimal cost. The second approach is to consider a random initial condition. If the variance of the distribution for the initial condition is unity, then the trace of the Riccati operator (its nuclear norm) is the optimal cost. Conditions are derived that guarantee the optimal cost as a continuous function of the actuator location. Since the space of feasible actuator locations is compact, continuity of the cost function guarantees the existence of a minimum.

Four numerical examples are included to demonstrate the different aspects of the theory. Three of the examples discuss a viscously damped beam with Euler-Bernouli beam equation and simply supported boundary conditions. The fourth example represents a diffusion equation with Neumann boundary conditions. The beam equation is

$$w_{tt} + c_d w_t + w_{xxxx} = b_r u(t), t \geq 0, 0 < x < 1, \tag{4.25}$$

with $z = \begin{bmatrix} w \\ w_t \end{bmatrix}$. The equation is put into the state space form using the eigenfunctions of the operator $\dfrac{\partial^4}{\partial x^4}$ with the simply supported boundary conditions. Different orders of approximation are considered and, for each one, a greater number of modes are included and the optimal actuator location is calculated. In the first example the state weight in the LQ-cost is chosen to be $\begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix}$. Although this choice leads to strong convergence of finite-dimensional Riccati operators to the infinite-dimensional Riccati operator, the optimal locations do not converge. This example shows that strong convergence is not enough and uniform convergence is required. In the second example the state weight is set to $C = \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}$. This choice only weighs the w and not w_t. The C operator is compact and the optimal actuator locations converge. The third example considers the deflection only at a single point. The C operator picks the value of w at the point 0.5 or in other words $Cz = w(0.5)$. The C operator is again compact and the optimal actuator locations converge.

The diffusion equation is also approximated by a sequence of approximations with increasing number of modes. In this example the C operator is chosen to be $\sqrt{1000}I$ and both the optimal cost and the corresponding optimal location converge as predicted by the theory.

In [8] the focus is on an algorithm for finding the optimal actuator location. In many situations the matrices that are involved in the calculations have large dimensions. For instance, in finite element approaches, the size of the state variable is the same as the number of mesh points. As more refined meshes are considered the size of the system grows. Additionally, as the number of actuators or sensors increase, the number of variables also increases. Therefore, an optimization scheme is needed to guide the search. In this paper, the problem is posed as one of convex optimization with a new formulation considering a discrete set of possible positions for the actuators. Suppose there are n positions and m actuators that need to be placed. Then there is a one-to-one mapping between the positions of the actuators and a binary string of length n with m ones and $n - m$ zeros where each one corresponds to a position that is filled with an actuator. The optimization algorithm works on the space of such binary strings and is illustrated using finite element models for a pinned beam and a plate. The performance of the new method is compared to a genetic algorithm. The results illustrate that for ten actuators on the beam and ten actuators on the plate the convex algorithm was faster and more accurate. The results are also experimentally verified using a cantilevered beam where two actuators are placed in the optimal positions and also in other configurations. The results show than the optimal positions are the most effective in suppressing vibrations.

Another method is based on the observability Gramian which is discussed in [16] and [4]. Let r correspond to the sensor location. The observability Gramian $W(r)$ is a matrix-valued function of r. In order to define a cost function, a single-valued function is needed. There are multiple ways to achieve this. Let λ_i be the eigenvalues of $W(r)$ and let λ_s, λ_l the smallest and the largest eigenvalues respectively. Different cost functions based on the Gramian are given below. The optimal location is r^*.

$$r^* = \arg\max_{r\in\Omega} \lambda_l(r), \tag{4.26}$$

$$r^* = \arg\max_{r\in\Omega} \ \det\left(W(r)\right) = \prod_1^n \lambda_i, \tag{4.27}$$

$$r^* = \arg\max_{r\in\Omega} \ \mathrm{tr}\left(W(r)\right) = \sum_1^n \lambda_i, \tag{4.28}$$

$$r^* = \arg\max_{r\in\Omega} \ \left(\sum_1^n \lambda_i\right)\left(\prod_1^n \lambda_i\right)^{\frac{1}{n}}, \tag{4.29}$$

$$r^* = \arg\max_{r\in\Omega} \lambda_s(r). \tag{4.30}$$

The smallest eigenvalue of the Gramian W can approach zero as higher number of modes are included for some systems such as the the damped beam. Figure 4.2 shows the log of the smallest eigenvalue vs. the number of modes for a damped beam. The model parameters are from [29] and the details of the model are explained in chapter five of the thesis. Because the infinite-dimensional system is not exactly observable, the smallest eigenvalue of the finite-dimensional approximation approaches zero as the approximation order increases.

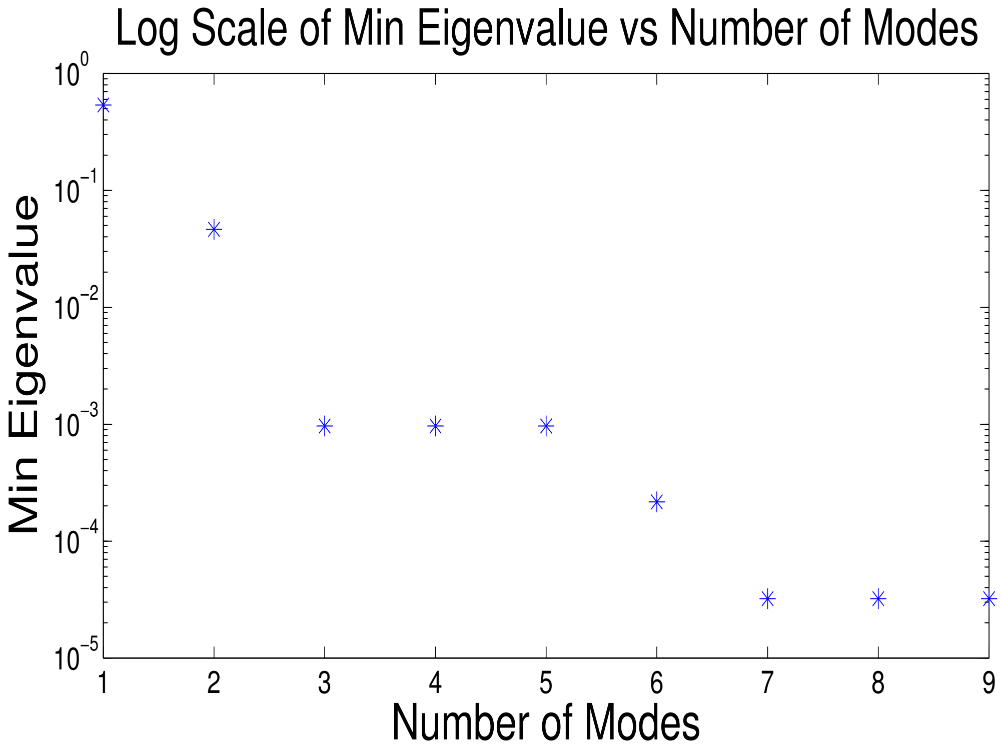


Figure 4.2: The figure shows the log of the minimum eigenvalue of the observability Gramian vs. the number of modes of a damped beam. The infinite-dimensional system is not observable therefore the minimum eigenvalue of the finite-dimensional approximations approach zero as more modes are considered.

The last method is from [26]. The strategy for placing the sensors is based on correlations between the outputs of sensors at different positions and performing multivariate regression analysis. The authors claim that this method does not require any explicit knowledge of the plant, however, it is assumed that chemical processes are governed by a specific PDE with a few free parameters that depend on the particular reaction and the reactor. The method is applied to two numerical models. Both of the models involve chemical reactions in tubular reactors. In the first model there are twelve possible positions for temperature sensors along the reactor. The goal is to use the temperature measurements to estimate the concentrations of different compounds. Utilizing regression analysis, two of the locations are selected as optimal. In contrast, the second model has nine sensor positions available with a goal to estimate the molar flow rate of different substances at the reactor outlet. It was discussed that two of the sensors are sufficient for optimal estimation. This approach for selecting sensor locations is not similar to the H_2 methods.

Chapter 5

Calculation of the Optimal Sensor Location

5.1 Simply Supported Beam

The Euler-Bernoulli (EB) beam equation describes the bending of a beam under the assumption that the cross-sections of the beam are rigid. For the derivation and a discussion of the assumptions in the EB equation see [3, chapter 5]. The left hand side of (5.1) is the EB equation and the right hand side contains additional disturbance and control terms. Equation (5.2) describes a single sensor measurement.

$$
\begin{aligned}
EI\frac{\partial^4 w(x,t)}{\partial x^4} + C_d\frac{\partial^5 w(x,t)}{\partial x^4 \partial t} + C_v\frac{\partial w(x,t)}{\partial t} + \mu\frac{\partial^2 w(x,t)}{\partial t^2}& \\
= b_1(x)d(t) + b_2(x)u(t), \quad 0 < x < L, 0 \leq t,& \qquad (5.1)\\
y(t) = \int_0^L c(x)w(x,t).& \qquad (5.2)
\end{aligned}
$$

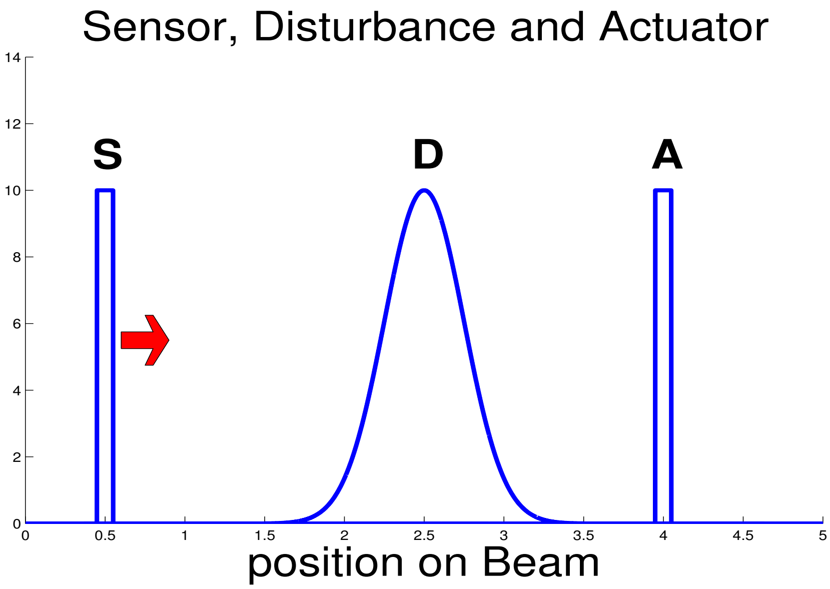


Figure 5.1: Functions S, D and A correspond to the sensor, the disturbance and the actuator and functions $c(x), b_1(x)$ and $b_2(x)$ respectively. The sensor is moved along the beam in order to find the optimal sensor location.

The parameters E, I, C_d, C_v, μ and L are the elastic modulus, moment of inertia of a cross-section, Kelvin-Voigt damping, viscous damping, mass density and the length of the beam. The parameter values are $EI = 4.91 \times 10^{-1}$ Kg m^3/s^2, $C_d = 6.49 \times 10^{-5}$Kg m^3/s, $C_v = 1.3 \times 10^{-3}$ Kg/ms, $\mu = 9.3$ Kg/m, $L = 5$m. The values are based on the experiments in [29] except the length L which is changed from $L = 0.45$ to $L = 5$ because the controlled performance in longer beams is more sensitive to sensor location.

The boundary conditions for a simply supported beam are

$$\begin{aligned} w(0,t) = 0, \quad w(L,t) &= 0, \\ EI\frac{\partial^2 w}{\partial x^2}(0,t) + C_d \frac{\partial^3 w}{\partial x^2 \partial t}(0,t) &= 0, \\ EI\frac{\partial^2 w}{\partial x^2}(L,t) + C_d \frac{\partial^3 w}{\partial x^2 \partial t}(L,t) &= 0. \end{aligned} \tag{5.3}$$

See [30, section 7.4] for a derivation of the boundary conditions. In the absence of damping, these boundary conditions reduce to

$$\begin{aligned} w(0,t) = 0, \quad & w(L,t) &= 0, \\ \frac{\partial^2 w}{\partial x^2}(0,t) = 0, \quad & \frac{\partial^2 w}{\partial x^2}(L,t) &= 0. \end{aligned} \tag{5.4}$$

Functions $b_1(x), b_2(x)$ and $c(x)$ are defined as (see Figure 5.1)

$$b_1(x, x_d) = 10e^{-(x-x_d)^2/\sigma^2} \tag{5.5}$$

$$b_2(x, x_a) = \begin{cases} 0 & |x - x_a| > d_a, \\ \dfrac{1}{2d_a} & |x - x_a| \leq d_a, \end{cases} \qquad c(x, x_s) = \begin{cases} 0 & |x - x_s| > d_s \\ \dfrac{1}{2d_s} & |x - x_s| \leq d_s. \end{cases} \tag{5.6}$$

Parameters x_d, x_a and x_s are the centers of the disturbance, the actuator and the sensor. They vary in the simulations. Parameters d_a and d_s correspond to half of the width of the sensor and the actuator while σ is the variance of the disturbance. Function b_1 has a factor of 10 so it has a comparable height with c and b_2. All of these values are modeling choices.

For each sensor location a generalized plant is created and the corresponding H_2-costs are calculated as defined in equations (4.5) and (4.6).

$$G(x_s) = \left[\begin{array}{c|cc} A & B_1 & B_2 \\ \hline C_1 & 0 & D_{12} \\ C_2(x_s) & D_{21} & 0 \end{array} \right]. \tag{5.7}$$

The process for creating each of the matrices in (5.7) is explained. Matrix A is found first.

To put the system in the standard state space form, a state variable is needed. Define $z(t) = \begin{bmatrix} w(x,t) \\ \frac{\partial}{\partial t} w(x,t) \end{bmatrix}$ as the state variable. A state space needs to be defined (see [24, Example 2.4]). Let $\mathcal{H}_2(0,L)$ be the Sobolev space defined as

$$\mathcal{H}_2(0,L) = \left\{ f(x) | \int_0^L f^2 + (f')^2 \, dx < \infty \right\}, \tag{5.8}$$

and $H_s(0,L) = \{f(x) \in \mathcal{H}_2(0,L) | f(0) = f(L) = 0\}$. Define the state space

$$Z = \begin{bmatrix} H_s(0,L) \\ \mathcal{L}_2(0,L) \end{bmatrix}, \qquad z(t) \in Z. \tag{5.9}$$

Rewrite the PDE (5.1) as

$$\dot{z}(t) = Az(t) + B_1 d(t) + B_2 u(t), \tag{5.10}$$

$$\dot{z}(t) = \frac{\partial}{\partial t} \begin{bmatrix} w(x,t) \\ \frac{\partial w(x,t)}{\partial t} \end{bmatrix} = A \begin{bmatrix} w(x,t) \\ \frac{\partial w(x,t)}{\partial t} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{\mu} b_1(x) \end{bmatrix} w(t) + \begin{bmatrix} 0 \\ \frac{1}{\mu} b_2(x) \end{bmatrix} u(t) \tag{5.11}$$

where the operator A is

$$A = \begin{bmatrix} 0 & I \\ -\frac{EI}{\mu} \frac{\partial^4}{\partial x^4} & -\frac{C_d}{\mu} \frac{\partial^4}{\partial x^4} - \frac{C_v}{\mu} I \end{bmatrix}. \tag{5.12}$$

The domain of the operator A is

$$D(A) = \left\{ \begin{bmatrix} \phi(x) \\ \psi(x) \end{bmatrix} | \psi \in \mathcal{H}_s(0,L), \right. \tag{5.13}$$

$$- \frac{EI}{\mu} \frac{\partial^4 \phi}{\partial x^4} - \frac{C_d}{\mu} \frac{\partial^4 \psi}{\partial x^4} - \frac{C_v}{\mu} \psi \in \mathcal{L}_2(0,L), \tag{5.14}$$

$$\phi(x) = 0, \quad x = 0, L \tag{5.15}$$

$$\left. EI \frac{\partial^2 \phi}{\partial x^2}(x) + C_d \frac{\partial^2 \psi}{\partial x^2}(x) = 0, \quad x = 0, L \right\} . \tag{5.16}$$

The solution of (5.1) is expanded in terms of the eigenfunctions of the undamped system because working with these eigenfunctions is easier than working with the eigenfunctions of the damped system. These correspond to the eigenfunctions of the matrix A in (5.12) with parameters $C_d = C_v = 0$. The following lemmas discuss the eigenfunctions of the operators $-\frac{EI}{\mu} \frac{\partial^4}{\partial x^4}$ and the undamped operator A.

Lemma 5.1.1. The eigenfunctions ϕ_n and the eigenvalues λ_n of the operator $\frac{\partial^4}{\partial x^4}$ in the interval $[0, L]$ with the boundary conditions $\phi(x) = \frac{\partial^2 \phi}{\partial x^2}(x) = 0$ at $x = 0,\ L$ are

$$\lambda_n = \left(\frac{n\pi}{L}\right)^4, \qquad \phi_n(x) = \sin\left(\frac{n\pi x}{L}\right). \tag{5.17}$$

Proof. By definition

$$\frac{\partial^4 \phi}{\partial x^4} = \lambda \phi, \tag{5.18}$$

$$\phi(x) = \frac{\partial^2 \phi}{\partial x^2}(x) = 0, \ \ x = 0, L. \tag{5.19}$$

Let w be a solution of $w^4 = \lambda$. Then $-w, iw, -iw$ are also solutions. Therefore

$$\phi(x) = c_1 e^{wx} + c_2 e^{-wx} + c_3 e^{iwx} + c_4 e^{-iwx}. \tag{5.20}$$

Using the four boundary conditions

$$c_1 + c_2 + c_3 + c_4 = 0, \tag{5.21}$$

$$c_1 e^{wL} + c_2 e^{-wL} + c_3 e^{iwL} + c_4 e^{-iwL} = 0, \tag{5.22}$$

$$c_1 + c_2 - c_3 - c_4 = 0, \tag{5.23}$$

$$c_1 e^{wL} + c_2 e^{-wL} - c_3 e^{iwL} - c_4 e^{-iwL} = 0. \tag{5.24}$$

In matrix notation

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ e^{wL} & e^{-wL} & e^{iwL} & e^{-iwL} \\ 1 & 1 & -1 & -1 \\ e^{wL} & e^{-wL} & -e^{iwL} & -e^{-iwL} \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}. \tag{5.25}$$

To get a non-zero solution the kernel of the first matrix on the left hand side must contain a nonzero element. Since adding or subtracting rows from each other does not change the kernel

$$\ker(\begin{bmatrix} 1 & 1 & 1 & 1 \\ e^{wL} & e^{-wL} & e^{iwL} & e^{-iwL} \\ 1 & 1 & -1 & -1 \\ e^{wL} & e^{-wL} & -e^{iwL} & -e^{-iwL} \end{bmatrix}) = \ker(\begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & e^{iwL} & e^{-iwL} \\ 1 & 1 & 0 & 0 \\ e^{wL} & e^{-wL} & 0 & 0 \end{bmatrix}). \tag{5.26}$$

In order for two of the rows to be dependent, w must be either $\dfrac{n\pi}{L}$ or $\dfrac{in\pi}{L}$. In either case

$$\lambda = w^4 \tag{5.27}$$
$$= \left(\frac{n\pi}{L}\right)^4. \tag{5.28}$$

If $w = \dfrac{n\pi}{L}$ then there are three independent rows in (5.26), the kernel has dimension one and

$$\begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ c_3 \\ -c_3 \end{bmatrix} \tag{5.29}$$

satisfies (5.25). Equations (5.29) and (5.20) imply

$$\phi(x) = ce^{iwx} - ce^{-iwx} \tag{5.30}$$
$$= c(\cos(wx) + i\sin(wx) - (\cos(-wx) - i\sin(-wx))) \tag{5.31}$$
$$= 2ci\sin(wx), \tag{5.32}$$
$$= 2ci\sin(\frac{n\pi x}{L}). \tag{5.33}$$

The factor $2ci$ can be dropped because a scaled eigenfunction is also an eigenfunction. A similar argument applies to the case $w = \dfrac{in\pi}{L}$. □

Lemma 5.1.2. The eigenfunctions and the eigenvalues of the operator $-\dfrac{EI}{\mu}\dfrac{\partial^4}{\partial x^4}$ with the same boundary conditions as in Lemma 5.1.1 are

$$\lambda_n = -\frac{EI}{\mu}\left(\frac{n\pi}{L}\right)^4, \qquad \phi_n(x) = \sin\left(\frac{n\pi x}{L}\right). \tag{5.34}$$

Proof. This follows immediately from Lemma 5.1.1 and (5.18). □

Lemma 5.1.3. The Eigenfunctions and eigenvalues of

$$A = \begin{bmatrix} 0 & I \\ -\frac{EI}{\mu}\frac{\partial^4}{\partial x^4} & 0 \end{bmatrix} \tag{5.35}$$

are

$$\lambda = \pm i\sqrt{\frac{EI}{\mu}}\left(\frac{n\pi}{L}\right)^2, \qquad \begin{bmatrix} \phi(x) \\ \psi(x) \end{bmatrix} = \begin{bmatrix} \sin(\frac{n\pi x}{L}) \\ \lambda\sin\left(\frac{n\pi x}{L}\right) \end{bmatrix}. \tag{5.36}$$

Proof. By definition

$$\lambda \begin{bmatrix} \phi(x) \\ \psi(x) \end{bmatrix} = A \begin{bmatrix} \phi(x) \\ \psi(x) \end{bmatrix} \tag{5.37}$$

$$= \begin{bmatrix} \psi(x) \\ -\frac{EI}{\mu}\frac{\partial^4 \phi(x)}{\partial x^4} \end{bmatrix}, \tag{5.38}$$

$$\phi(x) = \frac{\partial^2 \phi}{\partial x^2}(x) = 0, \;\; x = 0, L. \tag{5.39}$$

Therefore

$$\psi(x) = \lambda \phi(x), \tag{5.40}$$

$$-\frac{EI}{\mu}\frac{\partial^4 \phi(x)}{\partial x^4} = \lambda^2 \phi. \tag{5.41}$$

By Lemma, 5.1.2 $\lambda^2 = -\frac{EI}{\mu}\left(\frac{n\pi}{L}\right)^4$. Therefore,

$$\lambda = \pm i \sqrt{\frac{EI}{\mu}} \left(\frac{n\pi}{L}\right)^2. \tag{5.42}$$

Using Lemma 5.1.2 and (5.40)

$$\begin{bmatrix} \phi(x) \\ \psi(x) \end{bmatrix} = \begin{bmatrix} \sin(\frac{n\pi x}{L}) \\ \lambda \sin(\frac{n\pi x}{L}) \end{bmatrix}.$$

□

The deflection $w(x,t)$ is written as a sum of the normalized eigenfunctions. Rewrite

$$w(x,t) = \sum_{n=1}^{\infty} a_n(t) \sqrt{\frac{2}{L}} \sin\left(\frac{n\pi x}{L}\right). \tag{5.43}$$

The state variable $z(t)$ is created using the coefficients a_n and their time derivatives $\dot{a}_n$. For the first n coefficients of $w(t)$

$$z(t) = \begin{bmatrix} a_1(t) \\ a_2(t) \\ \vdots \\ a_n(t) \\ \dot{a}_1(t) \\ \dot{a}_2(t) \\ \vdots \\ \dot{a}_n(t) \end{bmatrix}. \tag{5.44}$$

Using (5.12), (5.43) and (5.44) the state matrix A of the size $2n \times 2n$ is

$$A = \left[\begin{array}{cccc|cccc}
0 & 0 & \cdots & 0 & 1 & 0 & \cdots & 0 \\
0 & 0 & \cdots & 0 & 0 & 1 & \cdots & 0 \\
0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\
0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 1 \\
\hline
-\frac{\pi^4 EI}{\mu L^4} & 0 & \cdots & 0 & -\frac{\pi^4 C_d}{\mu L^4} - \frac{C_v}{\mu} & 0 & \cdots & 0 \\
0 & -\frac{2^4\pi^4 EI}{\mu L^4} & \cdots & 0 & 0 & -\frac{2^4\pi^4 C_d}{\mu L^4} - \frac{C_v}{\mu} & \cdots & 0 \\
0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\
0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & 0 & \cdots & -\frac{N^4\pi^4 EI}{\mu L^4} & 0 & 0 & \cdots & -\frac{N^4\pi^4 C_d}{\mu L^4} - \frac{C_v}{\mu}
\end{array}\right]. \tag{5.45}$$

Matrices B_1, B_2 are described next. In (5.5), $b_1(x)$ is defined as a Gaussian function. Matrix B_1 is a $2N \times 2$ column vector where in the first column, the first half of the elements are zero because of (5.11) and the second half are the Fourier sine coefficients of $b_1(x)$.

$$b_1(x) = \sum_{n=1}^{\infty} b_{1n} \sqrt{\frac{2}{L}} \sin\left(\frac{n\pi}{L}x\right), \tag{5.46}$$

where

$$b_{1n} = \int_0^L b_1(x) \sqrt{\frac{2}{L}} \sin\left(\frac{n\pi}{L}x\right) dx. \tag{5.47}$$

The second column of B_1 is all zeros.

$$B_1 = \alpha \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \\ b_{11} & 0 \\ b_{12} & 0 \\ \vdots & \vdots \\ b_{1n} & 0 \end{bmatrix}. \tag{5.48}$$

The number of columns in B_1 correspond to the number of disturbances. There are two disturbances; one of them only affects the state and the other only affects the sensor. The effect of the disturbance on the sensor is described by D_{21} which is defined as

$$D_{21} = \begin{bmatrix} 0 & I \end{bmatrix}. \tag{5.49}$$

The structure for D_{21} and B_1 ensures that $D_{21}D_{21}^* = I$ and $B_1 D_{21}^* = 0$ which are taken as assumptions in section 3.1. These assumptions simply the algebraic derivations in the H_2-optimal theory. In the simulations B_1 is scaled by a scaling factor α for investigating the effect of the disturbance strength. The scaling factor varies between 1 and 10^3. For each result, the scaling factor is mentioned on the graph.

The matrix B_2 corresponds to the function $b_2(x)$ as defined in (5.6) and describes the actuator. It is a $2N \times 1$ column vector where the first half of its elements are zero because of (5.11) and the second half are the Fourier sine coefficients of $b_2(x)$ calculated according to

$$b_2(x) = \sum_{n=1}^{\infty} b_{2n} \sqrt{\frac{2}{L}} \sin\left(\frac{n\pi}{L}x\right) \tag{5.50}$$

where

$$\begin{aligned} b_{2n} &= \int_0^L b_2(x) \sqrt{\frac{2}{L}} \sin\left(\frac{n\pi}{L}x\right) dx \\ &= \frac{\sqrt{2L}}{2d_a n\pi} \left[\cos\left(\frac{n\pi}{L}(x_a - d_a)\right) - \cos\left(\frac{n\pi}{L}(x_a + d_a)\right)\right]. \end{aligned} \tag{5.51}$$

The matrix B_2 is

$$B_2 = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ b_{21} \\ b_{22} \\ \vdots \\ b_{2n} \end{bmatrix}. \tag{5.52}$$

Matrices C_1, D_{12} are defined as

$$C_1 = \begin{bmatrix} I_{n\times n} & 0_{n\times n} \\ 0_{1\times n} & 0_{1\times n} \end{bmatrix}_{n+1\times 2n}, \qquad D_{12} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}. \tag{5.53}$$

The above structure guarantees $C_1^* D_{12} = 0$ and $D_{12}^* D_{12} = I$ so that the assumptions A2 and A3 in section 3.1 are satisfied. These are simplifying assumptions in the H_2-optimal theory. The identity matrix in C_1 means all the coefficients in (5.43) are weighted equally. The zeros in the top right corner of C_1 mean that the second half of (5.44) is not included in the cost. These are the time derivative of the Fourier sine coefficients and correspond to the velocity of the beam. The matrix C_1 is scaled by different constants such as $10^0, 10^1, 10^2$ and 10^3 in the simulations to investigate the influence of the state weight. For each result, the corresponding scaling factor is mentioned on the same graph. Scaling C_1 does not change the orthogonality condition with D_{12}.

The matrix C_2 corresponds to the function $c(x)$ as defined in (5.6). The sensor measures the average of $w(x,t)$ in $[x_s - d_s, x_s + d_s]$. Similar to $b_1(x)$ and $b_2(x)$, the Fourier sine coefficients of $c(x)$, Equation (5.6), are found as

$$c(x) = \sum_{n=1}^{\infty} c_{2n} \sqrt{\frac{2}{L}} \sin\left(\frac{n\pi}{L} x\right), \tag{5.54}$$

where

$$\begin{aligned} c_{2n} &= \int_0^L c(x) \sqrt{\frac{2}{L}} \sin(\frac{n\pi}{L} x) dx \\ &= \frac{\sqrt{2L}}{2d_s n\pi} \left[\cos(\frac{n\pi}{L}(x_s - d_s)) - \cos(\frac{n\pi}{L}(x_s + d_s))\right]. \end{aligned} \tag{5.55}$$

Using (5.55),

$$C_2 = \begin{bmatrix} c_{21}, c_{22}, \ldots, c_{2n}, 0, 0, \ldots, 0 \end{bmatrix}. \tag{5.56}$$

Since there is only one sensor, C_2 has one row and because the sensor only measures w and not $\dot{w}$ the second half of C_2 is made of zeros. These zeros correspond to the time derivatives of the Fourier coefficients in (5.44).

In (5.52), (5.48), (5.45) and (5.44), only the first n coefficients of $w(x,t)$ are included. In [22], it is shown that for a damped beam the optimal actuator locations and the corresponding costs, calculated based on the nth system $(A_n, B_{1n}, B_{2n}, C_{1n})$, converge as n increases. Compactness of the operators B_1, B_2, C_2 and C_1 is relevant for the convergence of results. The first three operators are compact because either their input or output is finite-dimensional. Operator C_1, as defined in (5.53), is a compact operator between the spaces $H_s(0,L)$ and $\mathcal{L}_2(0,L)$. Convergence of the optimal sensor locations and the corresponding costs follows analogously to the results about actuators in [22].

The convergence is also shown numerically. Figures 5.2 and 5.3 show the effect of increasing n. The first figure shows the exact and approximate shapes of the disturbance and the actuator. The second figure shows that as n grows from 1 to 10 the graphs of the full control cost as defined in (4.5) converge. Parameters d_a and d_s are set to 0.05. This choice makes the sensor and the actuator localized. The variance of the disturbance, σ, is set to 0.25. This choice corresponds to a disturbance that has a wider width that the actuator.

In this section, the PDE is approximated using the Fourier sine series. If the PDE is approximated using a different approach, such as the local basis in the finite element method, one does not necessarily get the same system. As the approximation order increases, both methods converge to the original system, however, for low order approximations the costs will be different. On the other hand, if there exists a similarity transformation between two different systems, they both correspond to the same cost. In other words, costs J_1 and J_2 as defined in (4.5) and (4.6) are invariant under a similarity transformation.

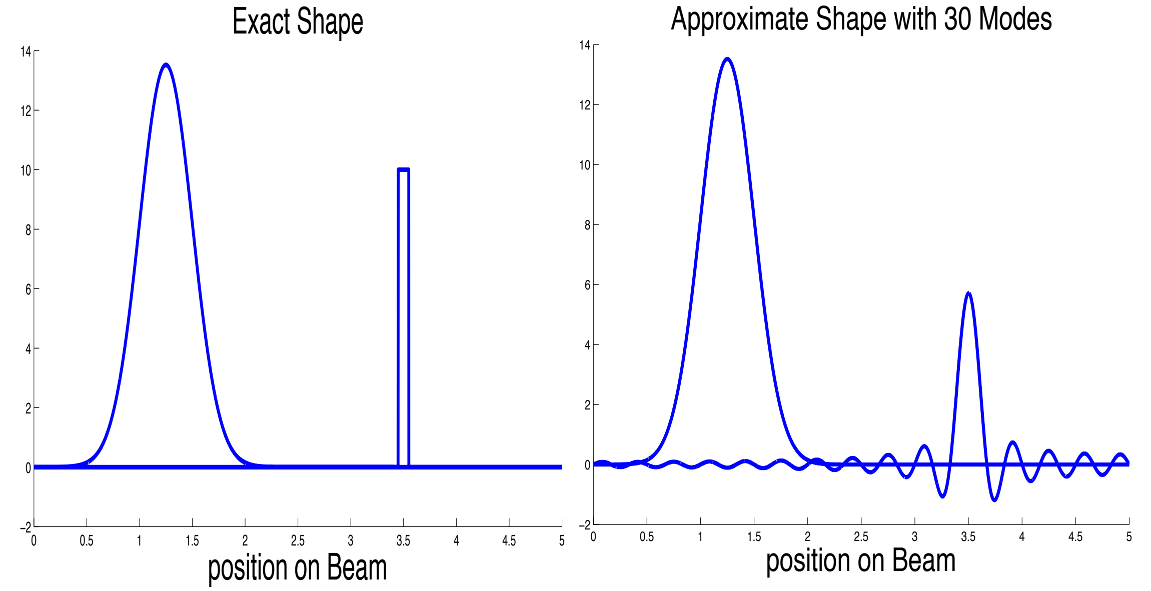
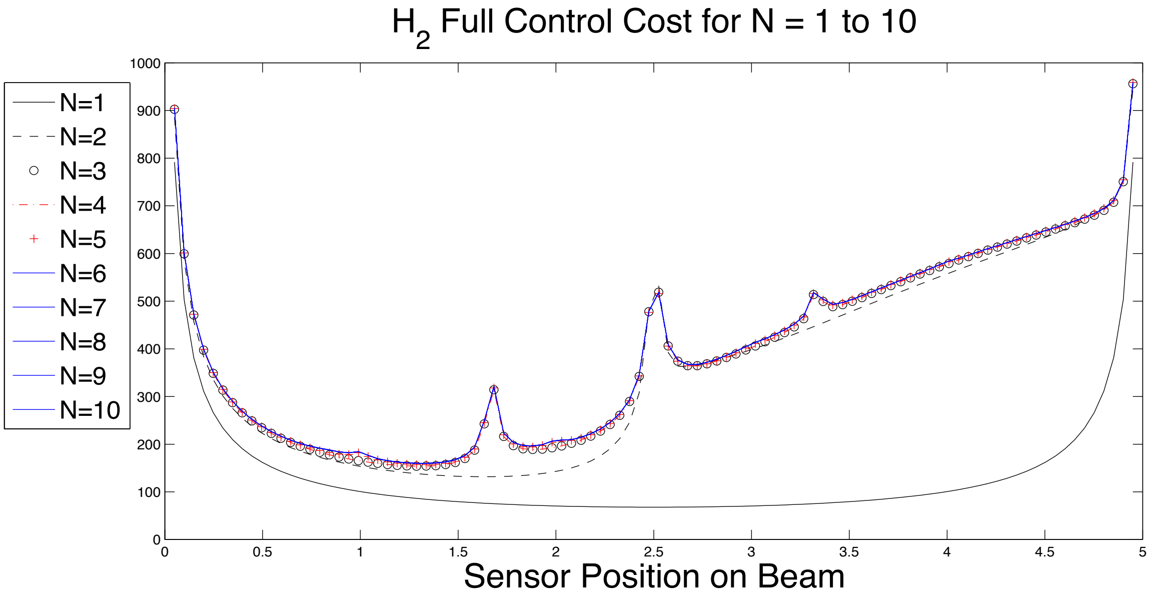
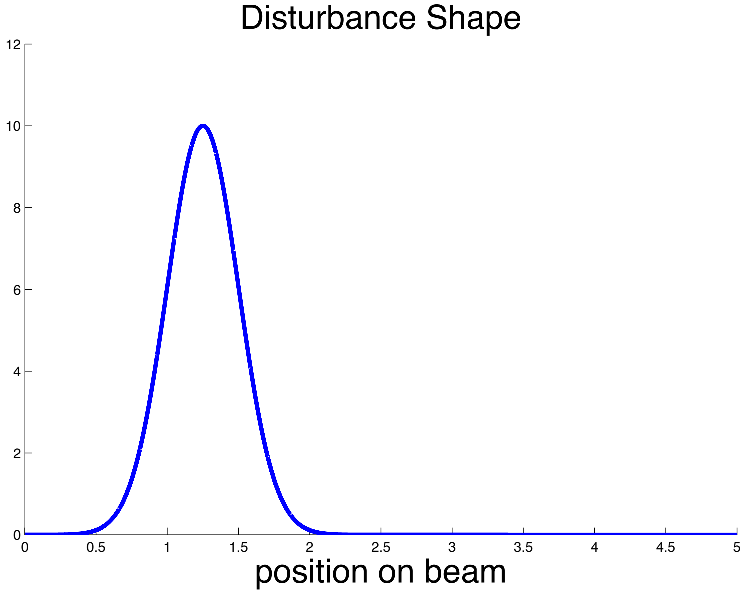


Figure 5.2: Exact shape of an interval characteristic function which is used as the actuator and sensor and a Gaussian function that represents the disturbance are shown in the left graph. In the right graph the approximations with 30 Fourier coefficients are shown. The approximations are used in the model.



(a)



(b)

N	$\|\|f_N - f_{N-1}\|\|_{\mathcal{L}_2}$
5	1722
6	393
7	16.84
8	0
9	0.43
10	0.23
11	0.032
12	0
13	0.0027
14	0.0014

Table 5.1: (c)

Figure 5.3: Figure(a) shows the full control cost for different number of modes. Variable N represents the number of modes and it ranges from 1 to 10. For $N = 6$ to 10 the graphs overlap. Figure(b) shows the shape of the disturbance. Matrix B_1 as defined in (5.48) is scaled by 3×10^2. This scaling moves the global minimum in Figure(a) towards the center of the Gaussian in Figure(b). Table(c) shows the L_2-norm of the difference between consecutive graphs in Figure(a). Function f_n represents the corresponding graph in Figure(a). Since the numbers in Table(c) are approaching zero, the corresponding functions f_n are converging.

The next series of figures describe the effect of changing different model parameters. In all the figures, the red cross is the optimal sensor location. Figure 5.4 shows a Gaussian disturbance on the left side of the beam, and the full control cost as a function of the sensor position. The energy of the disturbance, the norm of the first column of matrix B_1, is increased respectively by $10^0, 10^1, 10^2$ and 3×10^2. The series of plots show that as the scaling factor increases the optimal sensor location moves towards the center of the disturbance.

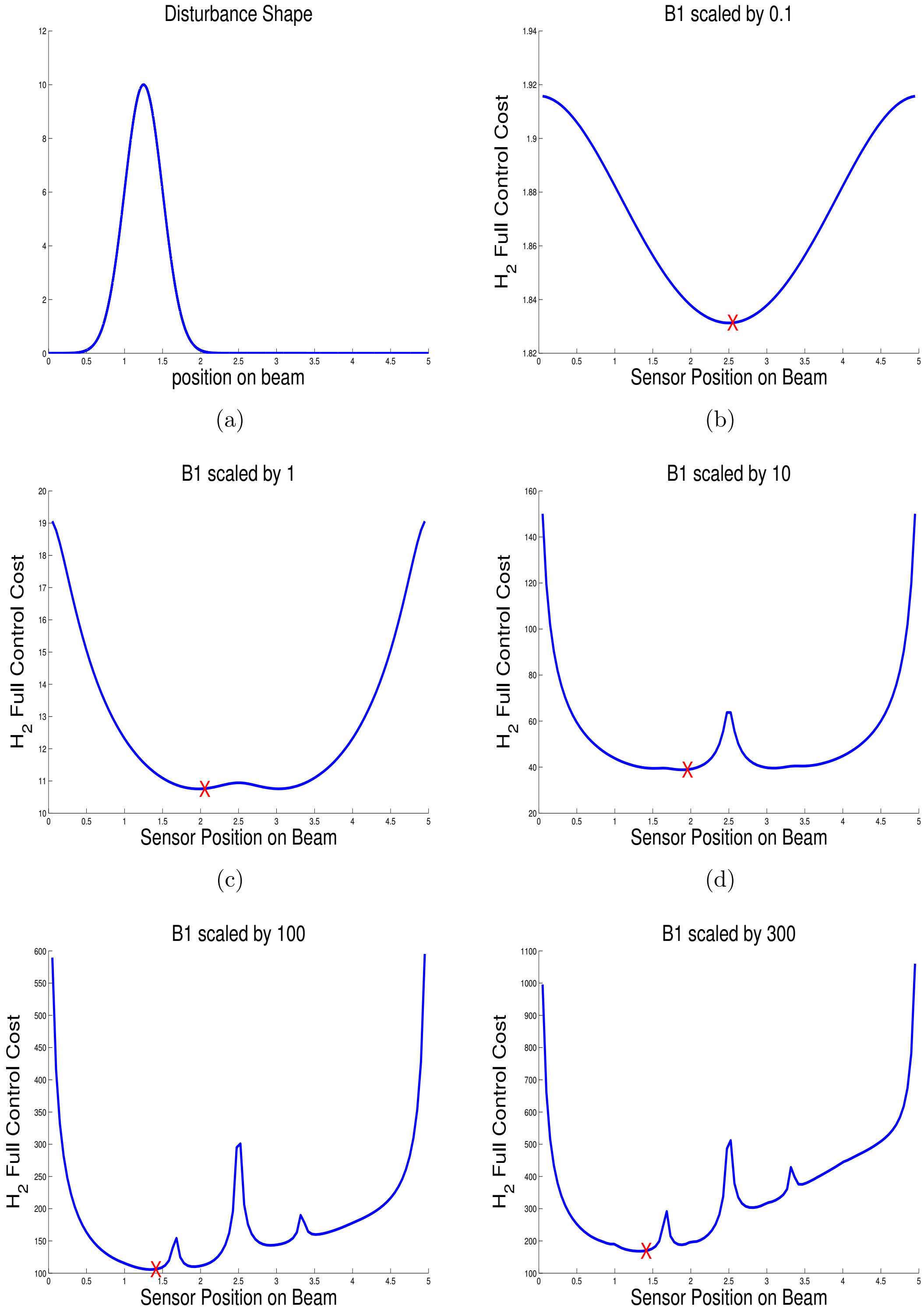
Figure 5.5 reaffirms the previous discussion by showing the disturbance in three different locations, while the scaling factor for B_1 is kept at 300. The three graphs show the full control cost vs. the sensor position on the beam, and they indicate, for different centers of disturbance, the optimal sensor location is close to the center of the disturbance.

Figure 5.6 shows the output feedback cost for three different combinations of the disturbance and actuator positions. As the actuator is moved to the right, the shape of the graph does not change significantly, however scaling B_2 does change the shape. In Figures 5.7 and 5.8, the affect of scaling B_2 is shown. In Figure 5.7, B_2 is scaled by 10^3 and, as the actuator moves, the optimal sensor location moves in the same direction. In Figure 5.8, B_2 is scaled by $10^0, 10^1, 10^2$ and 10^3. As the scaling factor increases, a location between the disturbance and the actuator corresponds to a cost that is close to the optimal cost.

The effect of the state weight, C_1, on the output feedback cost is shown in Figure 5.9. The disturbance and the actuator are fixed while, C_1 is scaled by $10^0, 10^1, 10^2$ and 10^3. The different scaling factors change the shape of the graph, but, they do not change the optimal sensor location significantly.

Finally, Figure 5.10 shows the effect of scaling B_1 in the output feedback cost. Matrices C_1 and B_2 are fixed while B_1 is scaled by $10^0, 10^1, 10^2, 3 \times 10^2$ and 10^3.

The results can be summarized as follows. Scaling matrices B_1 and C_1, with factors larger than one, moves the optimal sensor location towards the center of the disturbance for both full control and output feedback costs. Scaling B_2 creates another location, between the disturbance and the actuator, that has a cost close to the optimal value. In the full control case, if B_1 is scaled by a factor less than or equal to one, the optimal sensor location moves closer to the middle of the beam. The scaling factors for B_1, B_2 and C_1 change the matrices that are part of the Riccati equations in (4.5) and (4.6), and therefore, affect the optimal sensor location. In the absence of any scaling, the optimal location for the full control cost is off-centered towards the disturbance, and in the output feedback cost, it is at the center of the beam. The results illustrate that the optimal sensor location tends to be close to the disturbance. This is in agreement with the results in [10], where the placement optimization is done for two sensors, and the optimal location for one of the sensors is found to be at the disturbance location. Similarly, the analogous results for the optimal actuator location from [22] show that the optimal location is close to the disturbance.



(a)

(b)

(c)

(d)

(e)

(f)

Figure 5.4: Figure(a) shows the disturbance. The next five graphs show the H_2-full control cost vs sensor location with different scaling factors for the disturbance. The disturbance is represented by the first column of matrix B_1 in the state space equation, (5.48). As the scaling factor changes from 1 to 300, the optimal sensor location approaches the disturbance center. The red cross is the optimal sensor location.

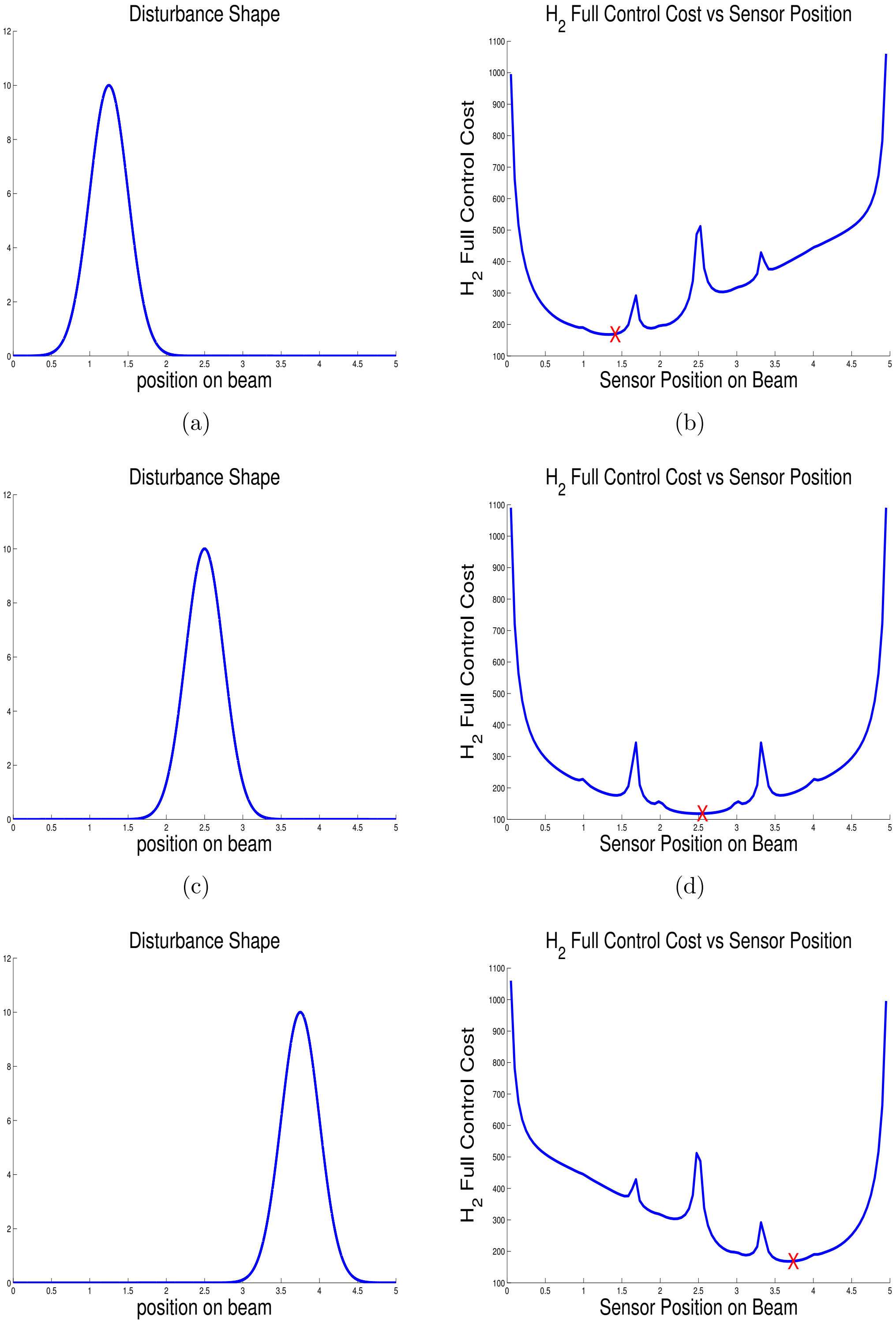


Figure 5.5: Figures(a,c,e) show the disturbance shape while Figures(b, d, f) show the full control cost. The length of the beam is divided into 100 points and for a sensor placed at each position the H_2 full control cost is calculated. In all of the simulations, B_1 as defined in (5.48) is scaled by 3×10^2. This scaling moves the global minimum of the figures on the right towards the centers of the Gaussian functions on the left.

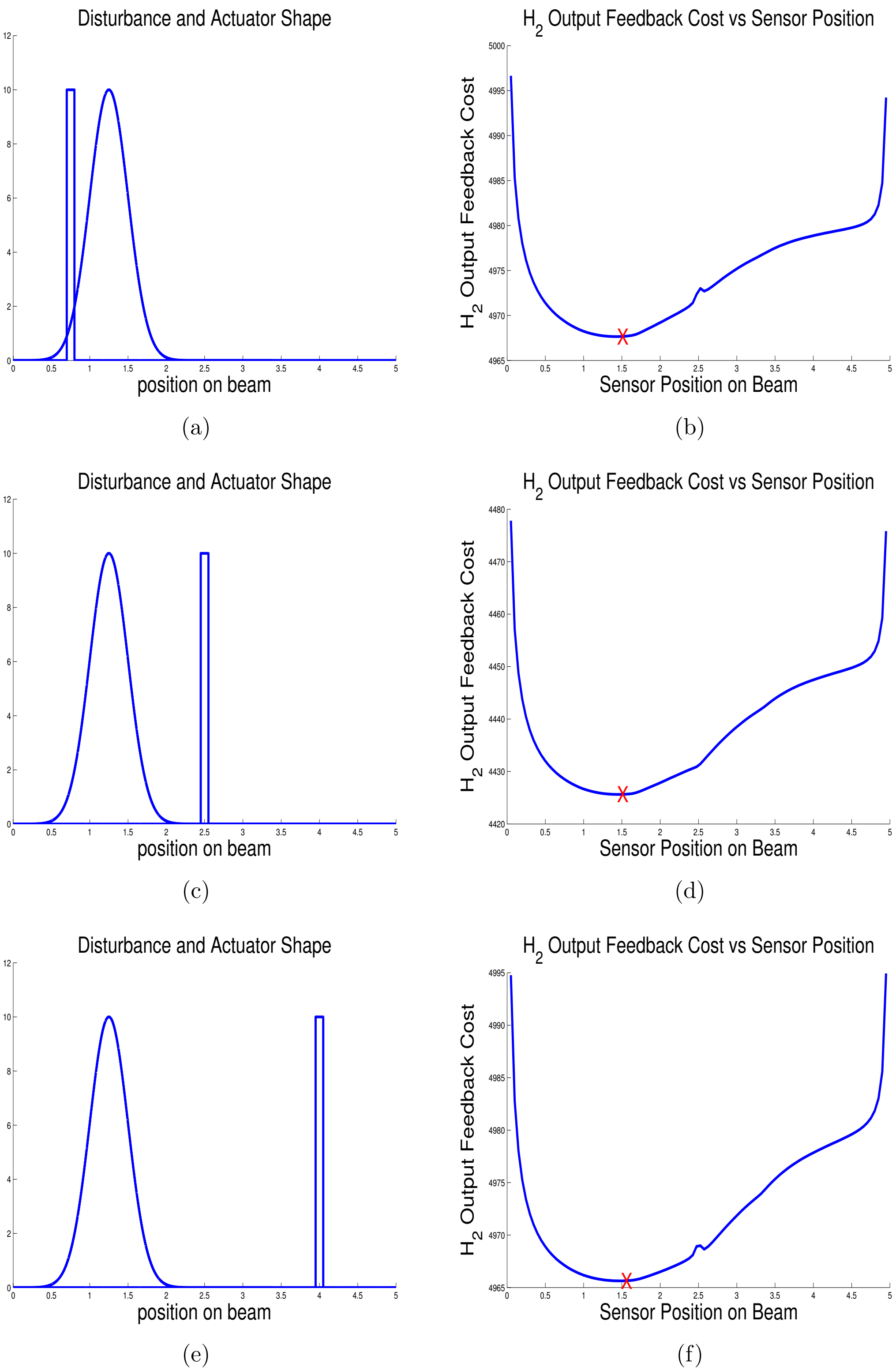
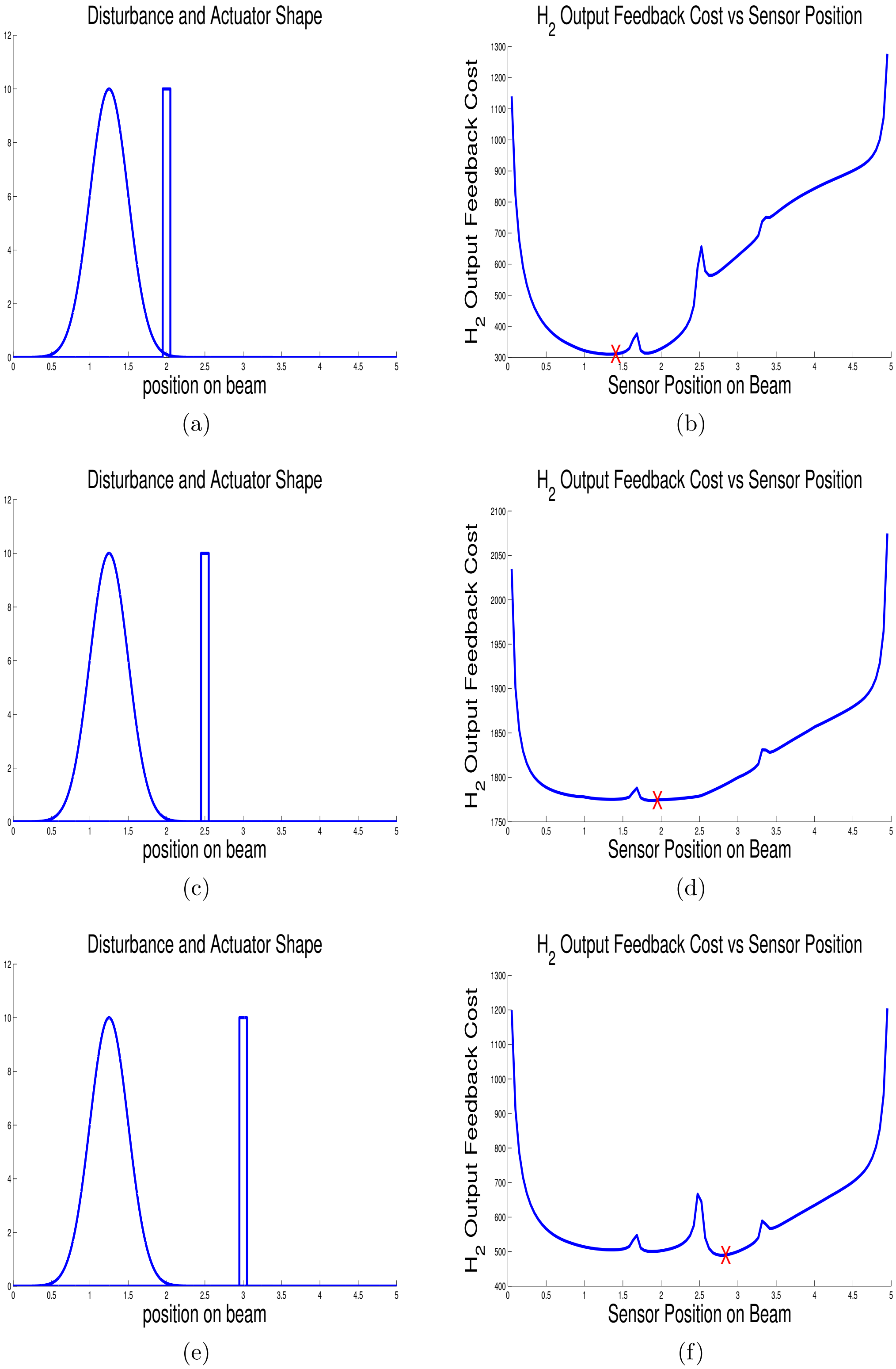


Figure 5.6: Figures(a,c,e) show the shape of the disturbance B_1 and the actuator B_2. The corresponding figures on the right show the output feedback cost. In the output feedback cost the disturbance and the actuator are involved. Matrix B_1 is scaled by 300 in all the three simulations. As the actuator moves to the right, the optimal sensor location does not change significantly. Matrix B_2 has a smaller influence compared to B_1. The red cross shows the optimal sensor location.



(a)

(b)

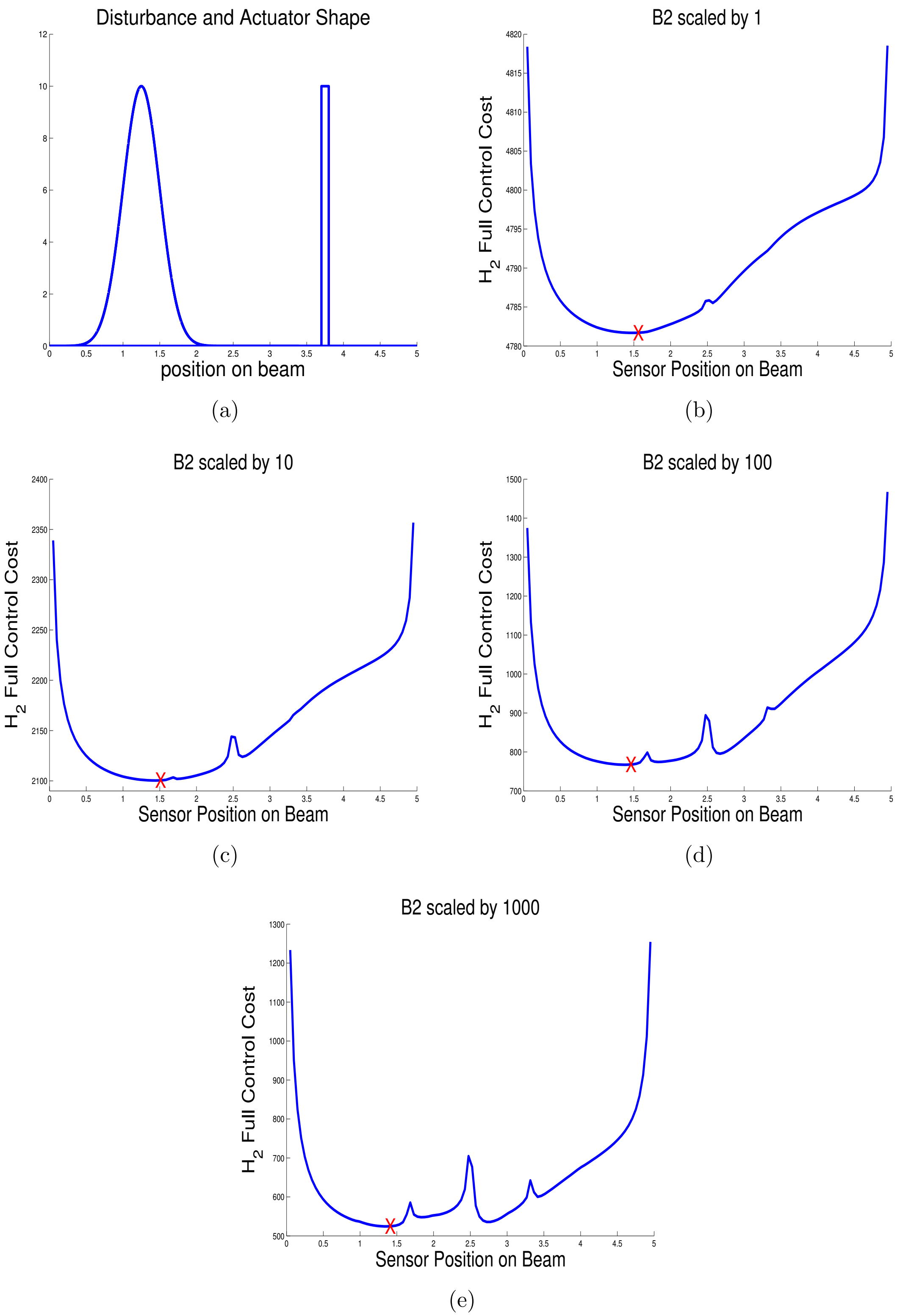
(c)

(d)

(e)

(f)

Figure 5.7: Figures(a,c,e) show the shapes of the disturbance B_1 and the actuator B_2. The corresponding figures on the right show the output feedback cost. Matrices B_1 and B_2 are scaled by 3×10^2 and 10^3 in the three simulations. As the actuator moves to the right, the optimal sensor location moves to right as well. In contrast to Figure 5.6, the actuator location affects the optimal sensor location. The red cross shows the optimal sensor location.



(a)

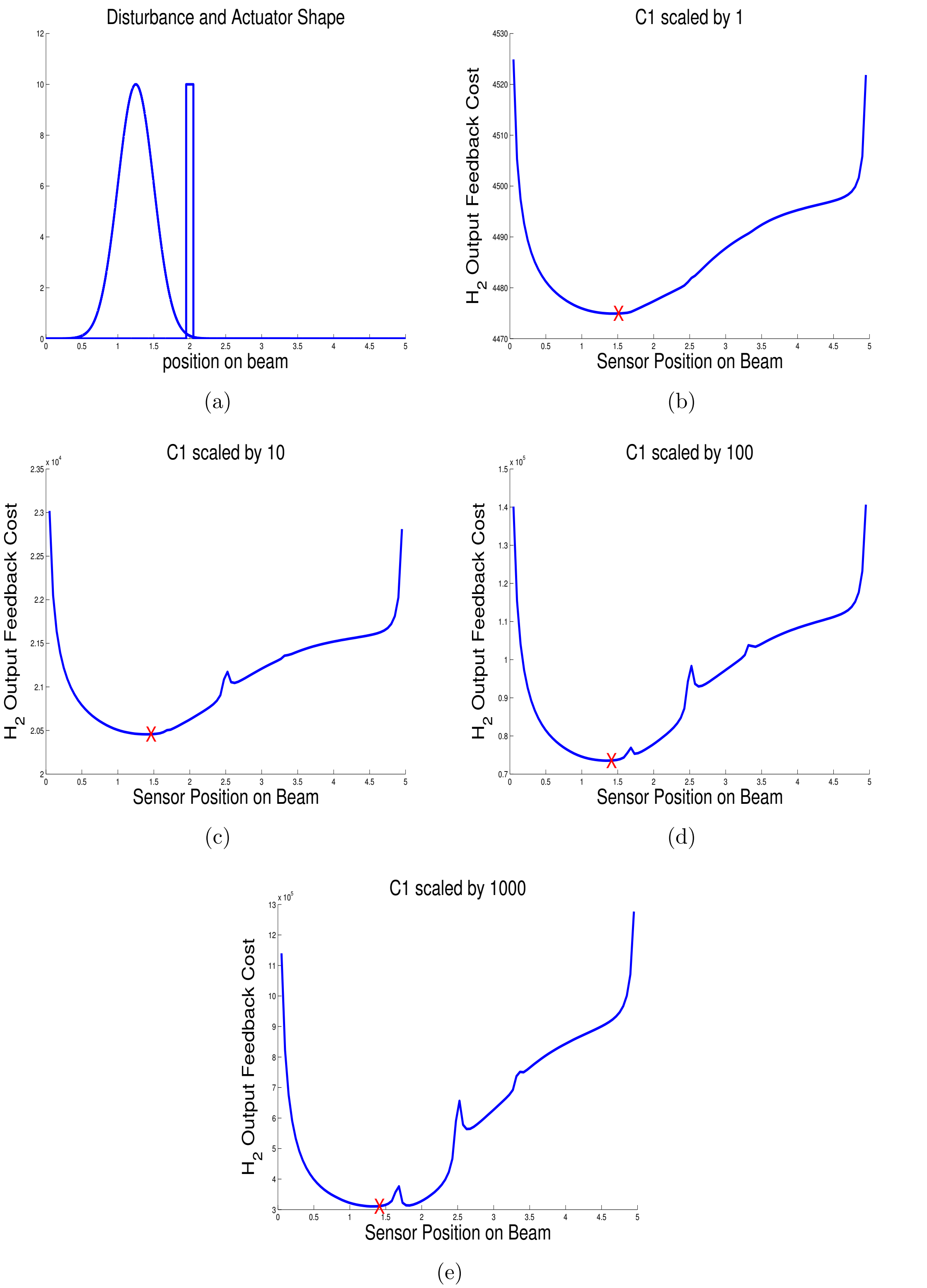
(b)

(c)

(d)

(e)

Figure 5.8: Figure(a) shows the disturbance B_1 and the actuator B_2. The output feedback cost is calculated for four different scaling factors for B_2. The scaling factors are $10^0, 10^1, 10^2$ and 10^3 and are indicated in the appropriate figures. The matrix B_1 is scaled by 300. The red cross is the optimal sensor location. Although scaling B_2 changes the shape of the graph for Figures(b,c,d) it does not significantly change the optimal sensor location. In Figure(e) the sensor location just to the right of the middle of the beam has a cost that is close to the optimal cost.



(a)

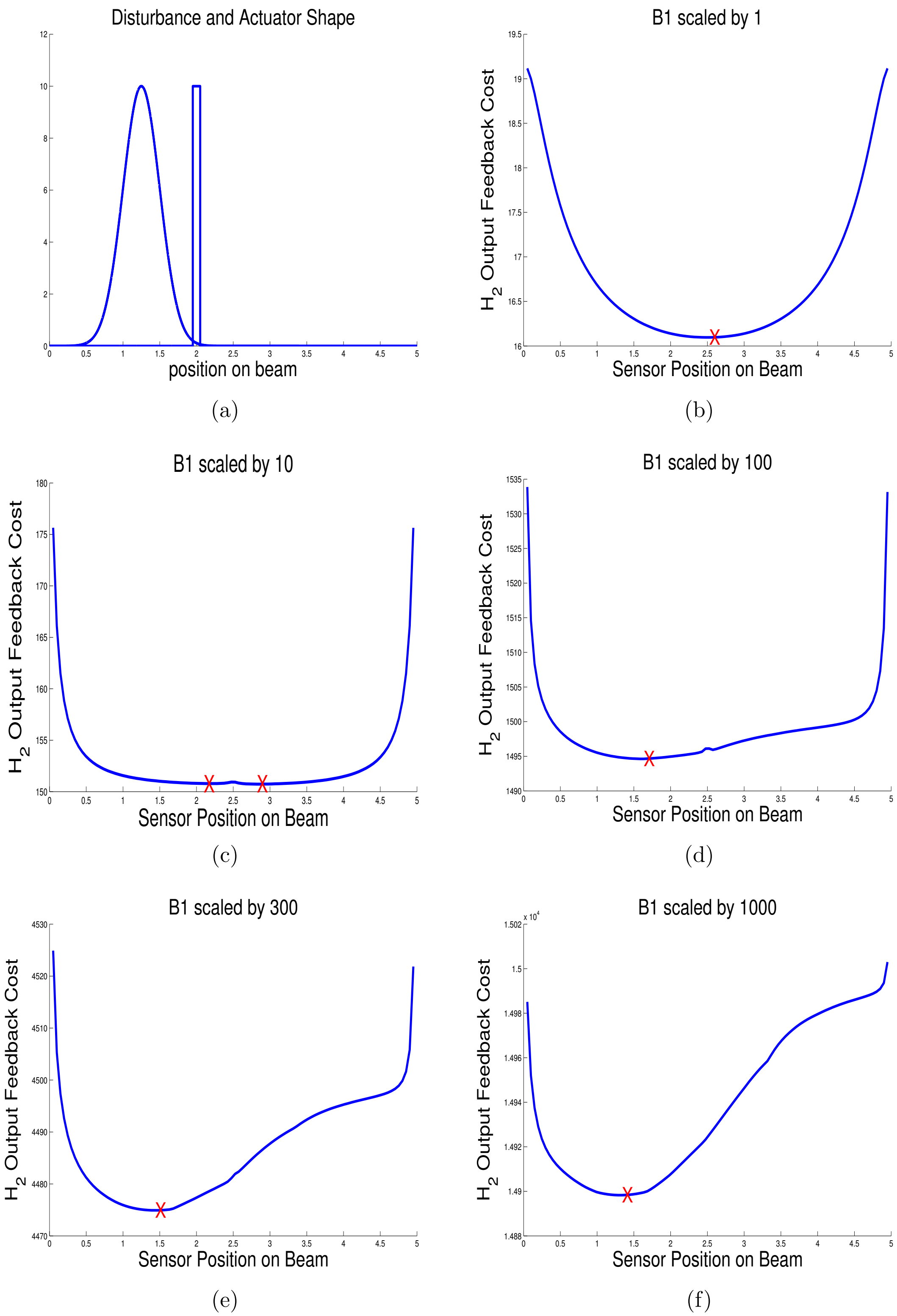
(b)

(c)

(d)

(e)

Figure 5.9: Figure(a) shows the shape of the disturbance B_1 and the actuator B_2. The other four figures show the output feedback cost for different scalings of C_1. The red cross shows the optimal sensor location. Increasing the scale factor for C_1 changes the graph of the H_2 output feedback cost but does not change the optimal sensor location significantly. Matrix B_1 is scaled by by 300.



(a)

(b)

(c)

(d)

(e)

(f)

Figure 5.10: Figure(a) shows the disturbance B_1 and actuator B_2. The other four figures show the output feedback cost for different scalings of B_1. Matrices C_1 and B_2 are not scaled. As the scaling factor increases the optimal sensor location moves toward center of the disturbance. Figure(c) is symmetric with respect to the middle of the beam, therefore there are two optimal sensor locations.

5.2 Diffusion in Two Dimensions

The first equation in (5.57) is the two-dimensional diffusion equation with additional disturbance and control terms. The second equation describes the sensor measurement and the third equation is the boundary condition. The domain is the L-shaped region shown in Figure 5.11 and is represented by Ω. The parameter σ describes the diffusivity.

$$\frac{\partial T(x,y,t)}{\partial t} = \sigma\left(\frac{\partial^2 T(x,y,t)}{\partial x^2} + \frac{\partial^2 T(x,y,t)}{\partial y^2}\right) + b_1(x,y)w(t) + b_2(x,y)u(t), \tag{5.57}$$

$$y(t) = \int_\Omega c(x,y)T(x,y,t)dxdy, \tag{5.58}$$

$$T(x,y,t) = 0 \quad (x,y) \in \partial\Omega. \tag{5.59}$$

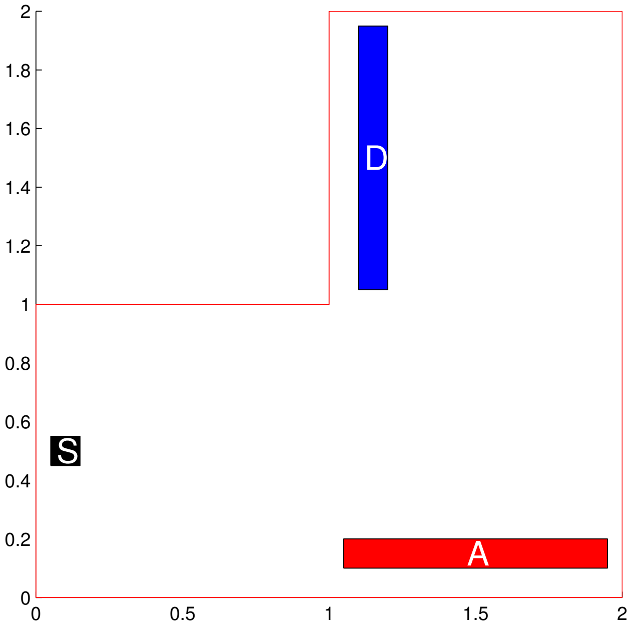


Figure 5.11: The domain is an L-shaped region. There are three regions inside the domain. The black region, labeled S, represents the sensor. The blue region, labeled D, represents the disturbance. The red region, labeled A, represents the actuator.

The disturbance, actuator and sensor functions are

$$b_1(x,y) = \begin{cases} 0 & (x,y) \notin D \\ \dfrac{1}{\text{area}(D)} & (x,y) \in D \end{cases}, \quad b_2(x,y) = \begin{cases} 0 & (x,y) \notin A \\ \dfrac{1}{\text{area}(A)} & (x,y) \in A \end{cases}, \tag{5.60}$$

$$c(x,y) = \begin{cases} 0 & (x,y) \notin S \\ \dfrac{1}{\text{area}(S)} & (x,y) \in S \end{cases}. \tag{5.61}$$

The state space is defined to be $\mathcal{L}_2(\Omega)$. The state operator is $A = \sigma\nabla^2$. Let $\mathcal{H}_2(\Omega)$ be a Sobolev space defined as

$$\mathcal{H}_2(\Omega) = \left\{ \int_\Omega f^2 + |\nabla f|^2 \, d\Omega < \infty \right\}. \tag{5.62}$$

The domain of the operator is defined as

$$D(A) = \{ f(x,y) | f(x,y) \in \mathcal{H}_2(\Omega), f(x,y) = 0 \text{ when } (x,y) \in \partial\Omega \}. \tag{5.63}$$

In order to find the optimal sensor location, the sensor is moved on a grid, see Figure 5.12, and for each point on the grid the H_2-optimal costs as defined in (4.5) and (4.6) are calculated.

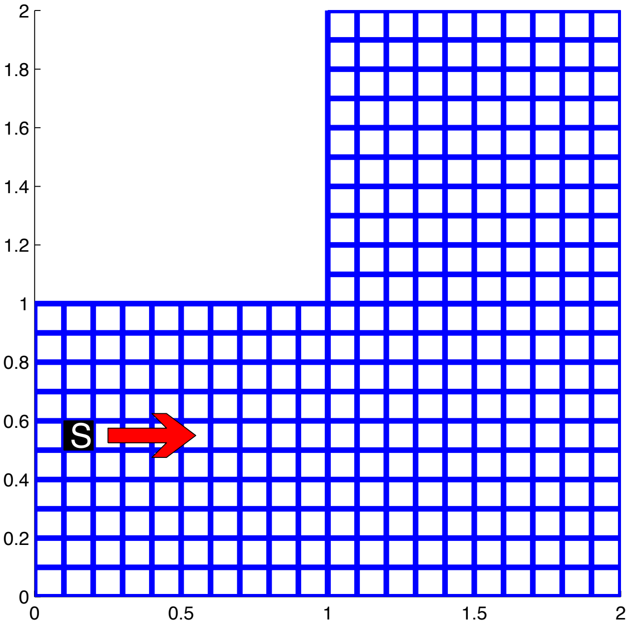


Figure 5.12: The cost function is calculated for each position of the sensor.

The PDE and the boundary conditions are projected onto a finite-dimensional state space using the Finite Element Method (FEM). For an introduction to FEM see [12], [17] or [27].

A mesh is required for FEM. The MATLAB commands in Appendix A code (1) create the L-shaped region and the mesh, see Figure 5.13.

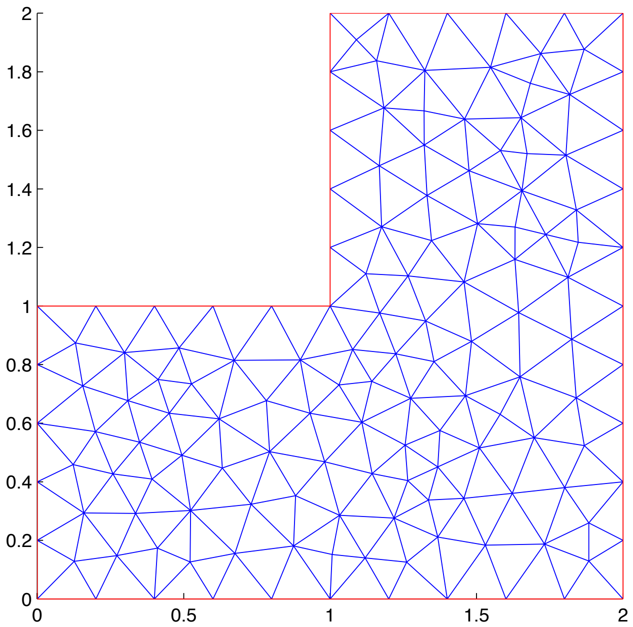
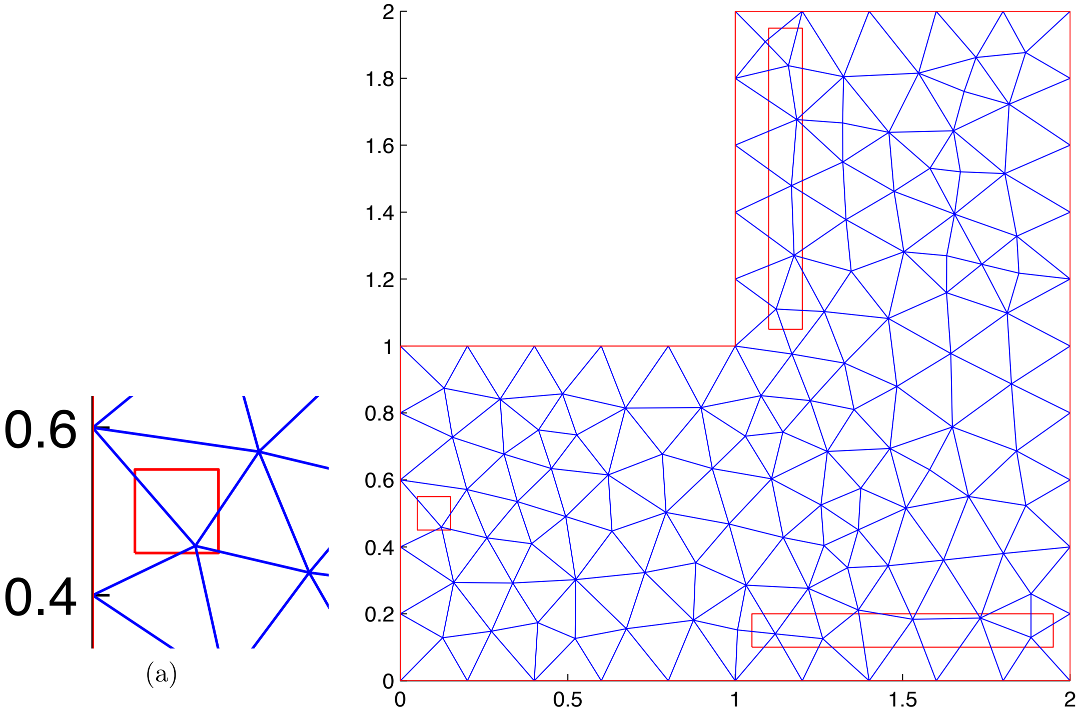


Figure 5.13: MATLAB commands decsg and initmesh create the L-shaped domain and a mesh respectively.

The superposition of the three regions S, D and A in Figure 5.11 and the mesh in Figure 5.13 results in partial overlap between the mesh triangles and the regions, see Figure 5.14. The partial overlap is an issue because in the generalized plant the sensor needs to be represented as a matrix and each vertex in the mesh corresponds to one of the elements of the state vector. For a triangle that partially overlaps with the sensor region, it is not clear if its vertices should be included in the sensor or not. There are several ways to handle this problem. For the function $c(x, y)$ and region S, the usual way is to project the function into the finite element space

$$[C_2]_j = \int_\Omega c(x, y)\phi_j(x, y)d\Omega. \tag{5.64}$$

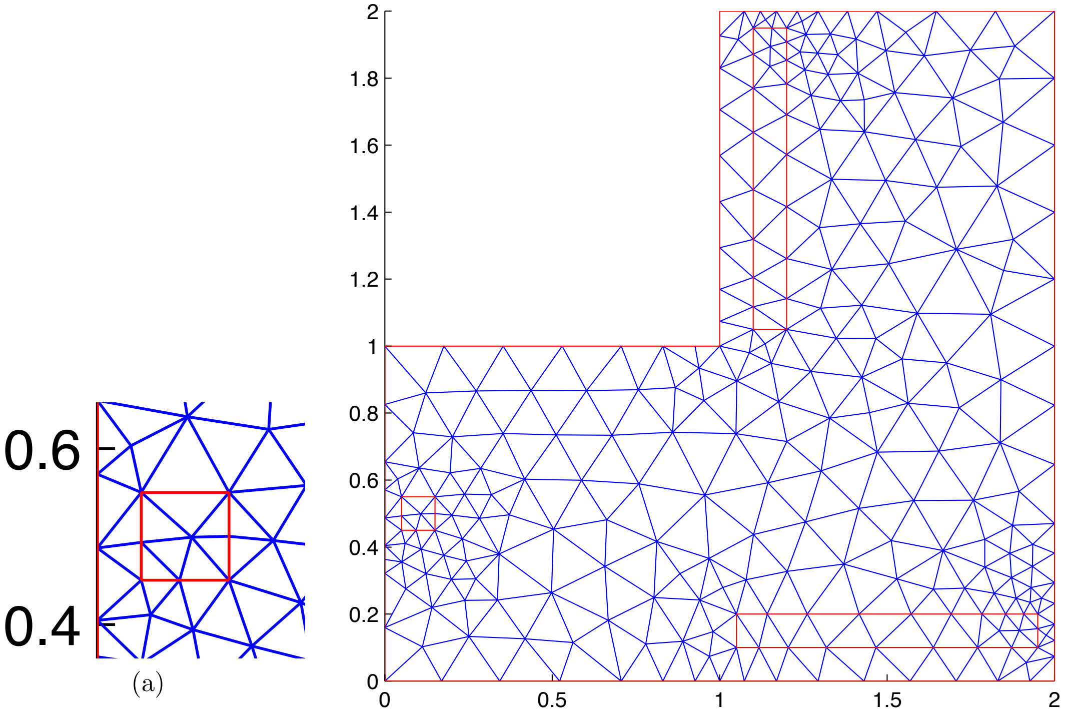
The mesh can be modified to reduce the partial overlap. Two approaches are discussed for modifying the mesh. The first way is to create the mesh in such a way that each triangle is either completely inside or outside each region, as in Figure 5.15. The MATLAB code in Appendix A code (2) creates a mesh such that the triangles fit the regions perfectly. The second way is to refine the mesh locally inside each region and only include the vertices that are inside the region, see Figure 5.16. The MATLAB code in Appendix A code (3) shows how to refine the mesh around the region S. The process for regions A and D is similar.



(a)

(b)

Figure 5.14: The three regions from Figure 5.11 and the mesh from Figure 5.13 are shown. Some of the mesh triangles partially overlap with the regions.



(a)

(b)

Figure 5.15: The triangles are either completely inside each region or outside. The MATLAB command initmesh considers the boundary of the regions when creating the mesh. Compared to Figure 5.14, there are additional small triangles around each region.

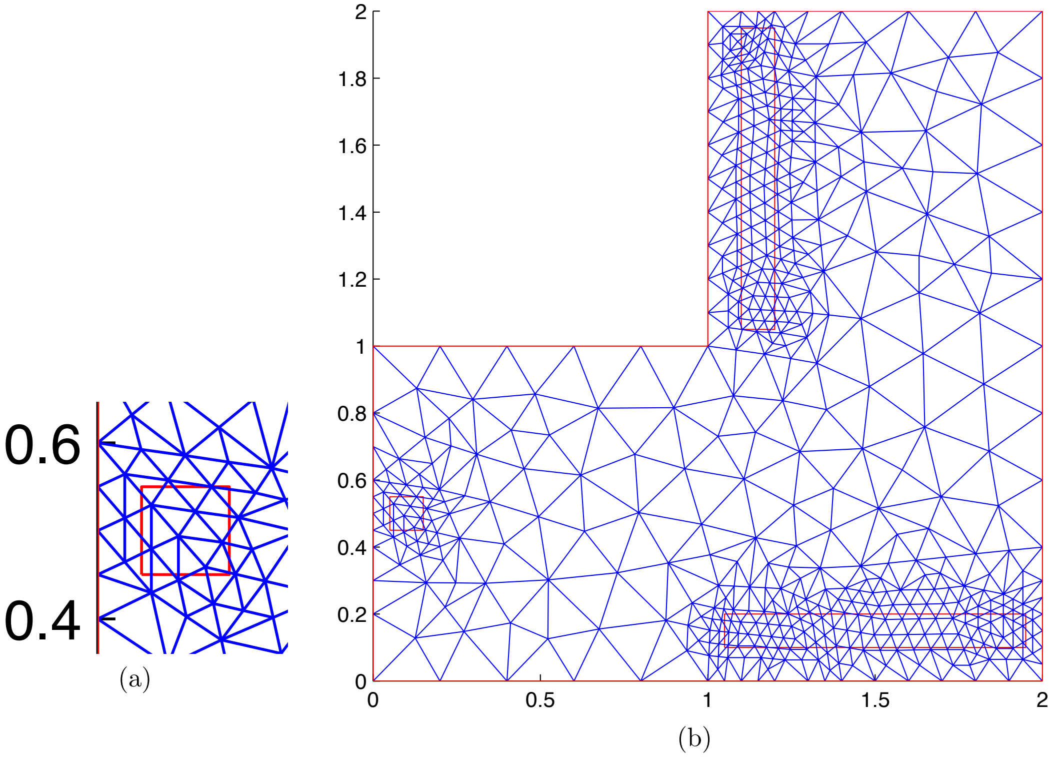
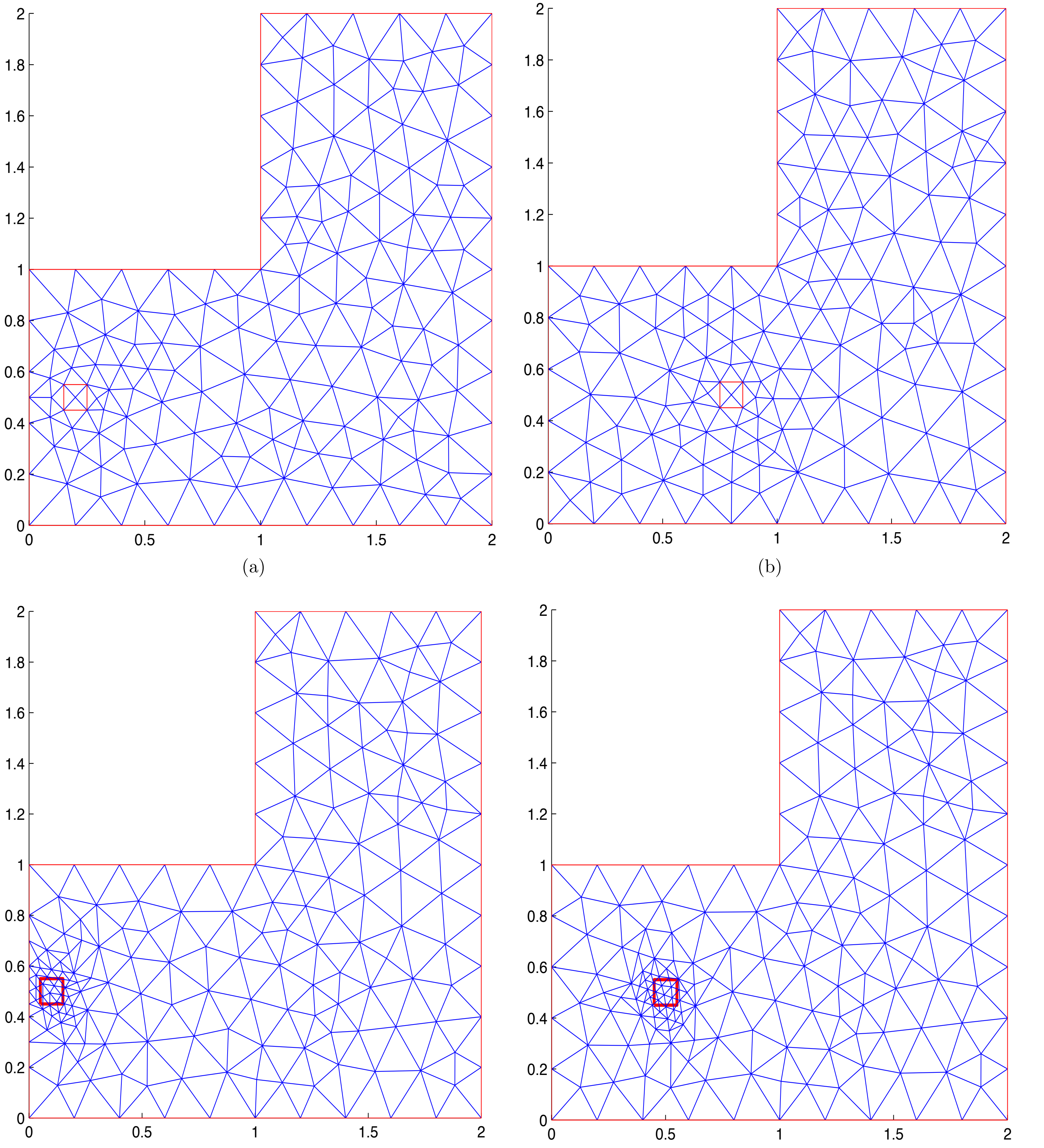


Figure 5.16: The mesh is locally refined inside each region.

As the sensor moves as shown in Figure 5.12, the mesh changes. If the mesh is organized according to the method in Figure 5.15 the entire mesh changes as the sensor moves. If the mesh is organized according to the method in Figure 5.16, the unrefined mesh stays fixed and some local refinement is added. These methods are computationally intensive for optimizing the sensor location because the mesh needs to be regenerated for every location of the sensor.



(a)

(b)

(c)

(d)

Figure 5.17: Figures a and b show the sensor at two different locations with meshes that are generated according to the method in Figure 5.15. Note that the entire mesh is adjusted. Figures c and d show the mesh refined locally at the sensor position as in Figure 5.16.

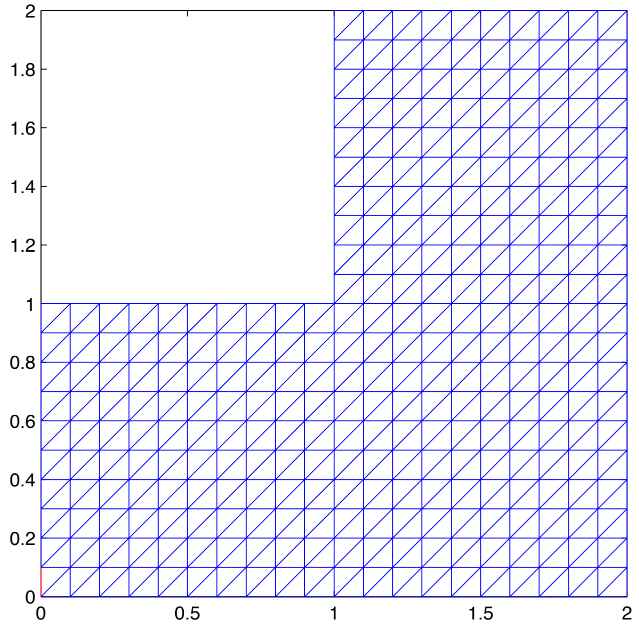


Figure 5.18: This type of mesh is called a structured mesh. As the sensor moves on the grid, the mesh does not change. However, the sensor must only move on the grid.

To avoid the mesh from changing when the sensor moves, a structured mesh can be used, see Figure 5.18. The advantage of this method is that the mesh is fixed. However, one disadvantage is that the sensor must move on the grid. This limits the step size for the sensor movement. Another disadvantage is that any refinement must apply to the entire domain to preserve the structure. The third disadvantage is that it is harder to cover the domains that are not shaped like a rectangle. In Figure 5.19 the results obtained by the structured mesh and the other two approaches are compared. Graphical comparison of the results suggests that the structured mesh is the best choice.

In order to obtain the space state form, equation (5.57) is multiplied by a smooth test function $V(x,y)$ that is zero on the boundary of the domain Ω and both sides of the equation are integrated over Ω. Using the Green's identity one of the spatial derivatives of T is passed to V. This is the standard procedure in FEM.

$$\iint\limits_{\Omega} VT_t = \sigma \iint\limits_{\Omega} V(T_{xx} + T_{yy}) + \iint\limits_{\Omega} Vb_1 w(t) + \iint\limits_{\Omega} Vb_2 u(t) \tag{5.65}$$

$$\iint\limits_{\Omega} VT_t = -\sigma \iint\limits_{\Omega} V_x T_x + V_y T_y + \oint_{\partial\Omega} V(T_x, T_y) \cdot n \, ds + \iint\limits_{\Omega} Vb_1 w(t) + \iint\limits_{\Omega} Vb_2 u(t). \tag{5.66}$$

Because V is zero on the boundary the second term on the right hand side is zero. Let $\phi_i(x, y)$ be the functions in the basis that is used to approximate the solution. They can be linear or higher order polynomials. Only linear terms are considered here because they provide sufficient accuracy. Approximate T, V, b_1 and b_2 by a linear combination of $\phi_i(x, y)$ as $T(x, y, t) = \sum_{i=1}^{n} \phi_i p_i(t)$, $V(x, y) = \sum_{i=1}^{n} \phi_i v_i$, $b_1(x, y) = \sum_{i=1}^{n} \phi_i b_{1i}$ and $b_2(x, y) = \sum_{i=1}^{n} \phi_i b_{2i}$. Using matrix notation T, V, b_1 and b_2 are rewritten as

$$T = \sum_{i=1}^{n} \phi_i p_i(t) = \left[\begin{array}{c} \phi_1, \phi_2, \ldots, \phi_{n-1}, \phi_n \end{array} \right] \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_{n-1} \\ p_n \end{bmatrix}, V = \sum_{i=1}^{n} \phi_i v_i = \left[\begin{array}{c} v_1, v_2, \ldots, v_{n-1}, v_n \end{array} \right] \begin{bmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_{n-1} \\ \phi_n \end{bmatrix}, \tag{5.67}$$

$$b_1 = \sum_{i=1}^{n} \phi_i b_{1i} = \left[\begin{array}{c} \phi_1, \phi_2, \ldots, \phi_{n-1}, \phi_n \end{array} \right] \begin{bmatrix} b_{11} \\ b_{12} \\ \vdots \\ b_{1(n-1)} \\ b_{1n} \end{bmatrix}, b_2 = \sum_{i=1}^{n} \phi_i b_{2i} = \left[\begin{array}{c} \phi_1, \phi_2, \ldots, \phi_{n-1}, \phi_n \end{array} \right] \begin{bmatrix} b_{21} \\ b_{22} \\ \vdots \\ b_{2(n-1)} \\ b_{2n} \end{bmatrix}. \tag{5.68}$$

Note that V has the transpose structure of the other three. Equation (5.66) is rewritten using the matrix notation.

$$\begin{aligned}
&\iint\limits_{\Omega} \begin{bmatrix} v_1, \ldots, v_n \end{bmatrix} \begin{bmatrix} \phi_1 \\ \vdots \\ \phi_n \end{bmatrix} \begin{bmatrix} \phi_1, \ldots, \phi_n \end{bmatrix} \begin{bmatrix} \dot{p_1} \\ \vdots \\ \dot{p_n} \end{bmatrix} = \\
&-\sigma \iint\limits_{\Omega} \begin{bmatrix} v_1, \ldots, v_n \end{bmatrix} \begin{bmatrix} \phi_{1x} \\ \vdots \\ \phi_{nx} \end{bmatrix} \begin{bmatrix} \phi_{1x}, \ldots, \phi_{nx} \end{bmatrix} \begin{bmatrix} p_1 \\ \vdots \\ p_n \end{bmatrix} - \sigma \iint\limits_{\Omega} \begin{bmatrix} v_1, \ldots, v_n \end{bmatrix} \begin{bmatrix} \phi_{1y} \\ \vdots \\ \phi_{ny} \end{bmatrix} \begin{bmatrix} \phi_{1y}, \ldots, \phi_{ny} \end{bmatrix} \begin{bmatrix} p_1 \\ \vdots \\ p_n \end{bmatrix} \\
&+ \iint\limits_{\Omega} \begin{bmatrix} v_1, \ldots, v_n \end{bmatrix} \begin{bmatrix} \phi_1 \\ \vdots \\ \phi_n \end{bmatrix} \begin{bmatrix} \phi_1, \ldots, \phi_n \end{bmatrix} \begin{bmatrix} b_{11} \\ \vdots \\ b_{1n} \end{bmatrix} + \iint\limits_{\Omega} \begin{bmatrix} v_1, \ldots, v_n \end{bmatrix} \begin{bmatrix} \phi_1 \\ \vdots \\ \phi_n \end{bmatrix} \begin{bmatrix} \phi_1, \ldots, \phi_n \end{bmatrix} \begin{bmatrix} b_{21} \\ \vdots \\ b_{2n} \end{bmatrix}.
\end{aligned} \tag{5.69}$$

Since $\begin{bmatrix} v_1, \ldots, v_n \end{bmatrix}$ is arbitrary, it is removed. Define the stiffness matrix K and the mass matrix M as

$$K = \sigma \iint\limits_{\Omega} \begin{bmatrix} \phi_{1x}\phi_{1x} + \phi_{1y}\phi_{1y} & \ldots & \phi_{1x}\phi_{nx} + \phi_{1y}\phi_{ny} \\ \vdots & \ldots & \vdots \\ \phi_{nx}\phi_{1x} + \phi_{ny}\phi_{1y} & \ldots & \phi_{nx}\phi_{nx} + \phi_{ny}\phi_{ny} \end{bmatrix}, \tag{5.70}$$

$$M = \iint\limits_{\Omega} \begin{bmatrix} \phi_1\phi_1 & \ldots & \phi_1\phi_n \\ \vdots & \ldots & \vdots \\ \phi_n\phi_1 & \ldots & \phi_n\phi_n \end{bmatrix}. \tag{5.71}$$

The mass matrix M is invertible. This is because each ϕ_i corresponds to a distinct row of M, and hence these rows can not be linearly dependent. This is equivalent to arguing that the non-overlapping mesh structure implies M is invertible. Equation (5.69) can be rewritten as

$$M \begin{bmatrix} \dot{p_1} \\ \vdots \\ \dot{p_n} \end{bmatrix} = -K \begin{bmatrix} p_1 \\ \vdots \\ p_n \end{bmatrix} + M \begin{bmatrix} b_{11} \\ \vdots \\ b_{1n} \end{bmatrix} w(t) + M \begin{bmatrix} b_{21} \\ \vdots \\ b_{2n} \end{bmatrix} u(t), \tag{5.72}$$

$$\begin{bmatrix} \dot{p_1} \\ \vdots \\ \dot{p_n} \end{bmatrix} = -M^{-1}K \begin{bmatrix} p_1 \\ \vdots \\ p_n \end{bmatrix} + \begin{bmatrix} b_{11} \\ \vdots \\ b_{1n} \end{bmatrix} w(t) + \begin{bmatrix} b_{21} \\ \vdots \\ b_{2n} \end{bmatrix} u(t). \tag{5.73}$$

Equation (5.73) is in the state space form. By matching (5.73) to (2.1), the following definitions are obtained.

$$A = -M^{-1}K, \quad \text{state variable} = \begin{bmatrix} p_1 \\ \vdots \\ p_n \end{bmatrix}, \quad B_1 = \begin{bmatrix} b_{11} & 0 \\ \vdots & \vdots \\ b_{1n} & 0 \end{bmatrix}, \quad B_2 = \begin{bmatrix} b_{21} \\ \vdots \\ b_{2n} \end{bmatrix}. \tag{5.74}$$

The additional column of zeros in B_1 correspond to a second disturbance that does not affect the state. This disturbance only affects the sensor measurement. The matrix D_{21} describes the effect of the disturbance on the sensor measurement. It is defined as $D_{21} = \begin{bmatrix} 0 & 1 \end{bmatrix}$. The structure for B_1 and D_{21} satisfies $B_1 D_{21}^* = 0$ and $D_{21}D_{21}^* = I$ which are simplifying assumptions from section 3.1. These are simplifying assumptions in H_2-control theory. For each b_{1i}, if the corresponding mesh vertex is inside the disturbance region, b_{1i} is one, otherwise it is zero. Similarly for b_{2i} if the corresponding mesh vertex is inside the actuator region, b_{2i} is one, otherwise it is zero. Then the first columns of B_1 and B_2 are normalized.

The MATLAB command "assema" was used to create the matrices M and K. The process for calculating the integrals in equations (5.70) and (5.71) is called assembling, hence the name "assema". See MATLAB's documentation for "assema".

```
[K,M,~]=assema(p,t,σ,1,0);
```

Recall that a generalized plant has the structure

$$G = \left[\begin{array}{c|cc} A & B_1 & B_2 \\ \hline C_1 & 0 & D_{12} \\ C_2 & D_{21} & 0 \end{array}\right].$$

The matrices C_1, C_2 and D_{12} are

$$C_1 = \begin{bmatrix} I_{n\times n} \\ 0_{1\times n} \end{bmatrix}_{n+1\times n}, C_2 = \begin{bmatrix} c_{21} & c_{22} & \dots & c_{2n} \end{bmatrix}, D_{12} \quad = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}_{n+1\times 1}. \tag{5.75}$$

The above structure guarantees $C_1^* D_{12} = 0$ and $D_{12}^* D_{12} = I$ which are simplifying assumptions from section 3.1. Scaling C_1 does not change these relationships. The identity matrix in C_1 means all the elements in the state variable are weighted equally. Since there is only one sensor, C_2 has one row. Each c_{2i} in C_2 corresponds to a mesh vertex. If the vertex with index i is inside the sensor region, c_{2i} is a constant otherwise C_{2i} is zero. The constant is set to a value so that C_2 is normalized to one.

In order to calculate the H_2-optimal costs (4.5) and (4.6), at least one Riccati equations needs to be solved for each position of the sensor. Solving the Riccati equations repetitively takes the majority of the calculation time. The MATLAB command for continuous time Riccati equations, “care”, solves the equation using Schur factorization of the Hamiltonian that is associated with the Riccati equation.

Figure 5.19 shows the the full control results for the three different types of mesh discussed previously. By graphical comparison of the results, it seems the structured mesh is the best choice. However if enough computational power is available, the results should converge for all three different mesh types given a refined enough mesh (see the comments on timing below). Additional artifacts can be introduced because of the contouring algorithm. See the MATLAB documentation for “contourf”.

An important issue is the convergence of the results when the mesh is refined. The results in [22] show that the optimal actuator costs and the optimal actuator locations converge for a sequence of finite element approximations with increasingly refined meshes. The convergence of optimal sensor costs and optimal sensor locations follow from the duality between the actuators and sensors. Figure 5.20 shows the full control costs on three structured meshes with resolutions, 0.2, 0.1, 0.05. As the mesh is refined the computational cost for solving the Riccati equations increases. The time for the simulations are roughly 10 seconds, 3 minutes and 10 hours respectively on a Mac mini with a 2.5 GHz Intel Core i5 processor. As the mesh element is divided by two in both x and y direction the number of mesh points is roughly multiplied by four. The size of matrix A in the Riccati equation quadruples in each direction and the number of Riccati equations that need to be solved increases by a factor of four. Because the computational cost can become impractical, typically an optimization procedure is used to guide the search. However since only one sensor is considered here no optimization method is used.

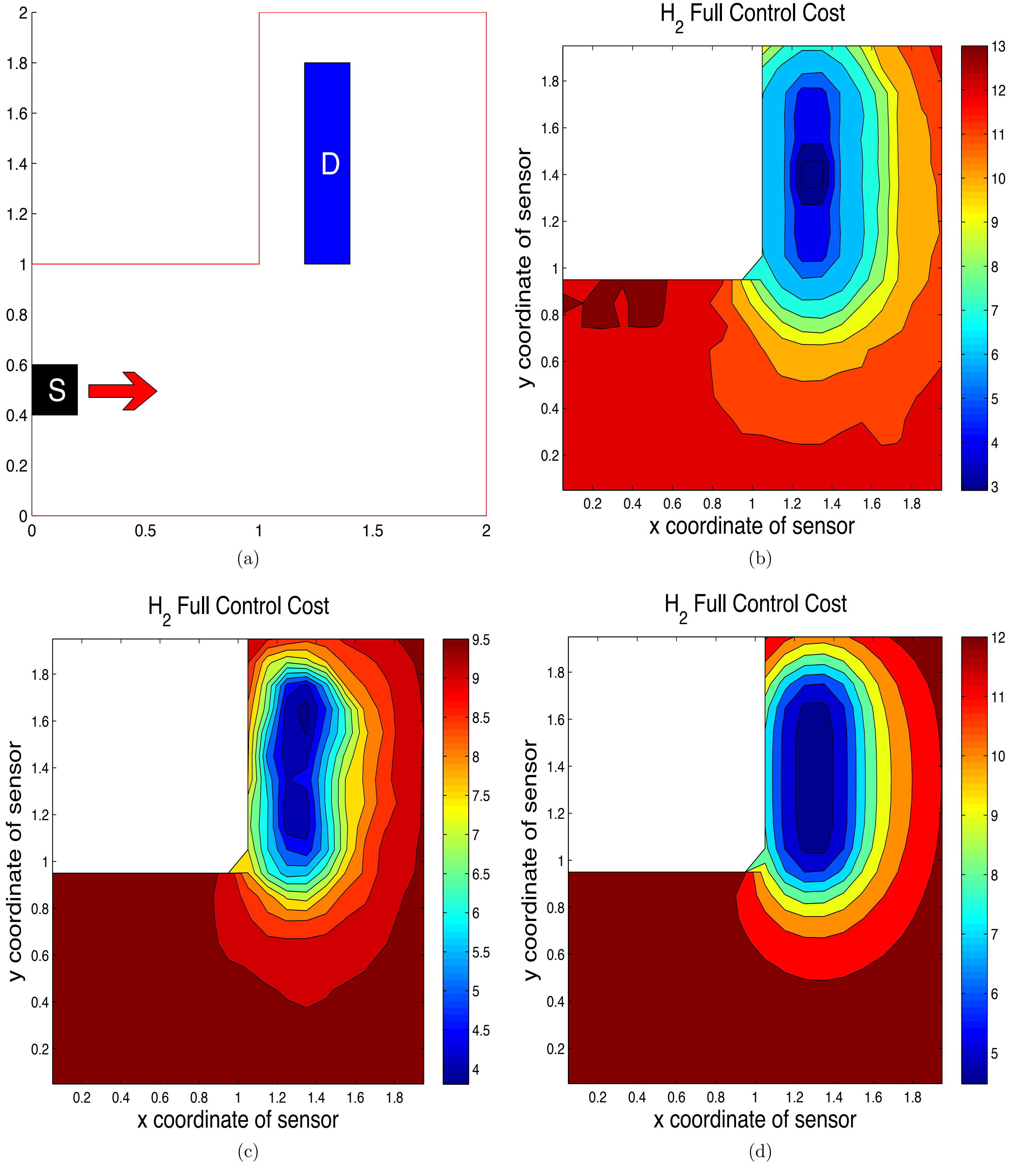
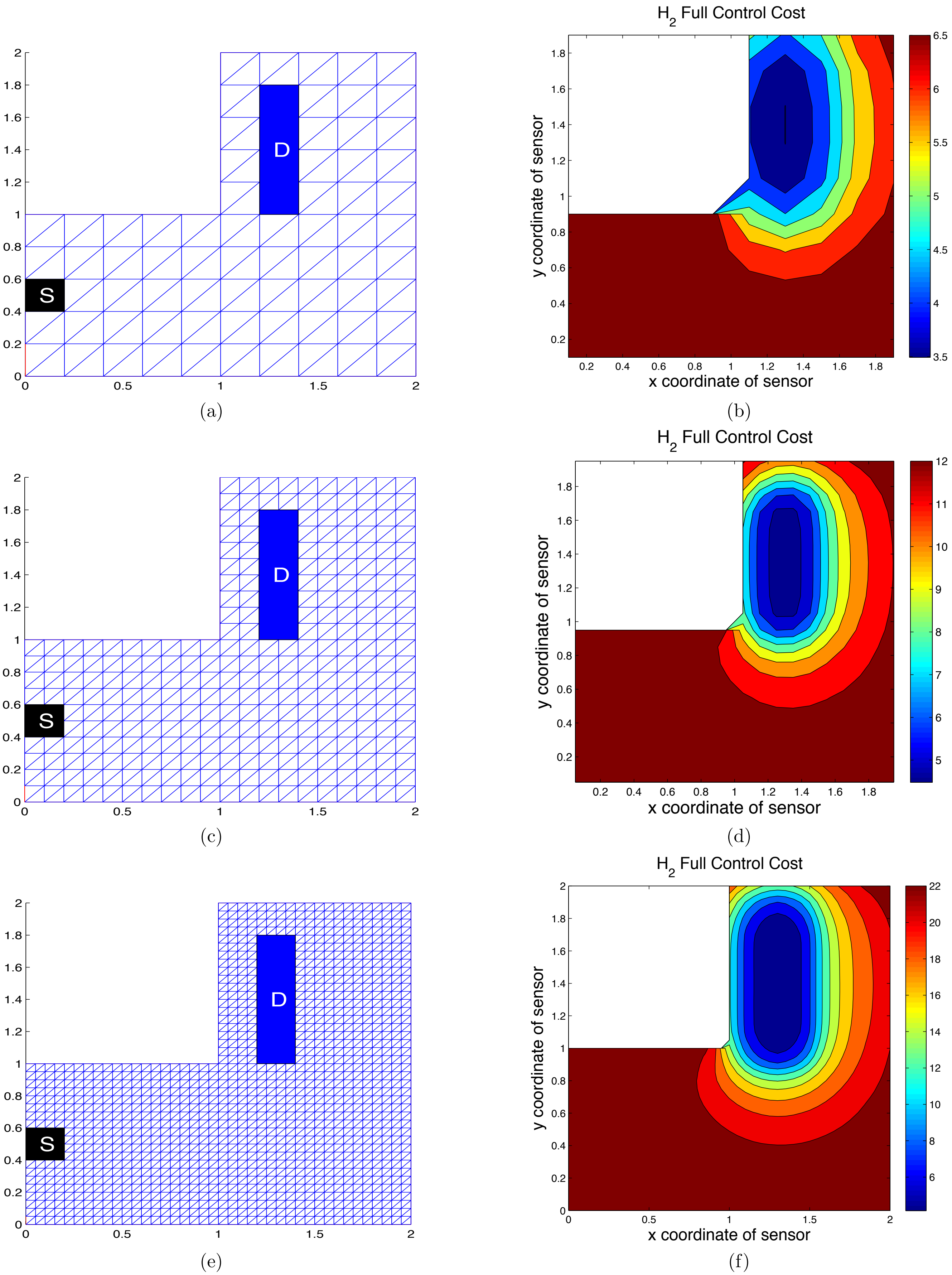


Figure 5.19: The full control results from the three different mesh types are shown. Figure b is based on the mesh in Figure 5.15 where mesh points are placed exactly on the boundary of the regions S and D. Figure c is based on the mesh from Figure 5.16 where there is a fixed mother mesh and the mesh is locally refined in the regions. Figure d is based on a structured mesh. Graphically comparing the results, the structured mesh seems to be the best choice. The resolution for the sensor movement in all three figures is 0.1 in both directions. Matrix B_1 is scaled by 10 and $\sigma = 1$.



(a) (b) (c) (d) (e) (f)

Figure 5.20: Figures a, c and e show three structured meshes with resolutions 0.2, 0.1 and 0.05. Figures b, d and f show the corresponding results. The largest values are 6.5, 12 and 22 and the lowest are 3.5, 4.5 and 4.2 respectively. Matrix B_1 is scaled by 10 and $\sigma = 1$.

The mesh comparisons show that additional artifacts can appear in the results if the mesh changes as the sensor moves. Using a structured mesh can solve this problem at the expense of more computation. Refining a structured mesh a few times can make the time for the computations impractical (on a desktop computer).

Based on comparing the results for the structured mesh on three different resolutions, the contours of constant cost value for the cost function become more smooth with mesh refinement.

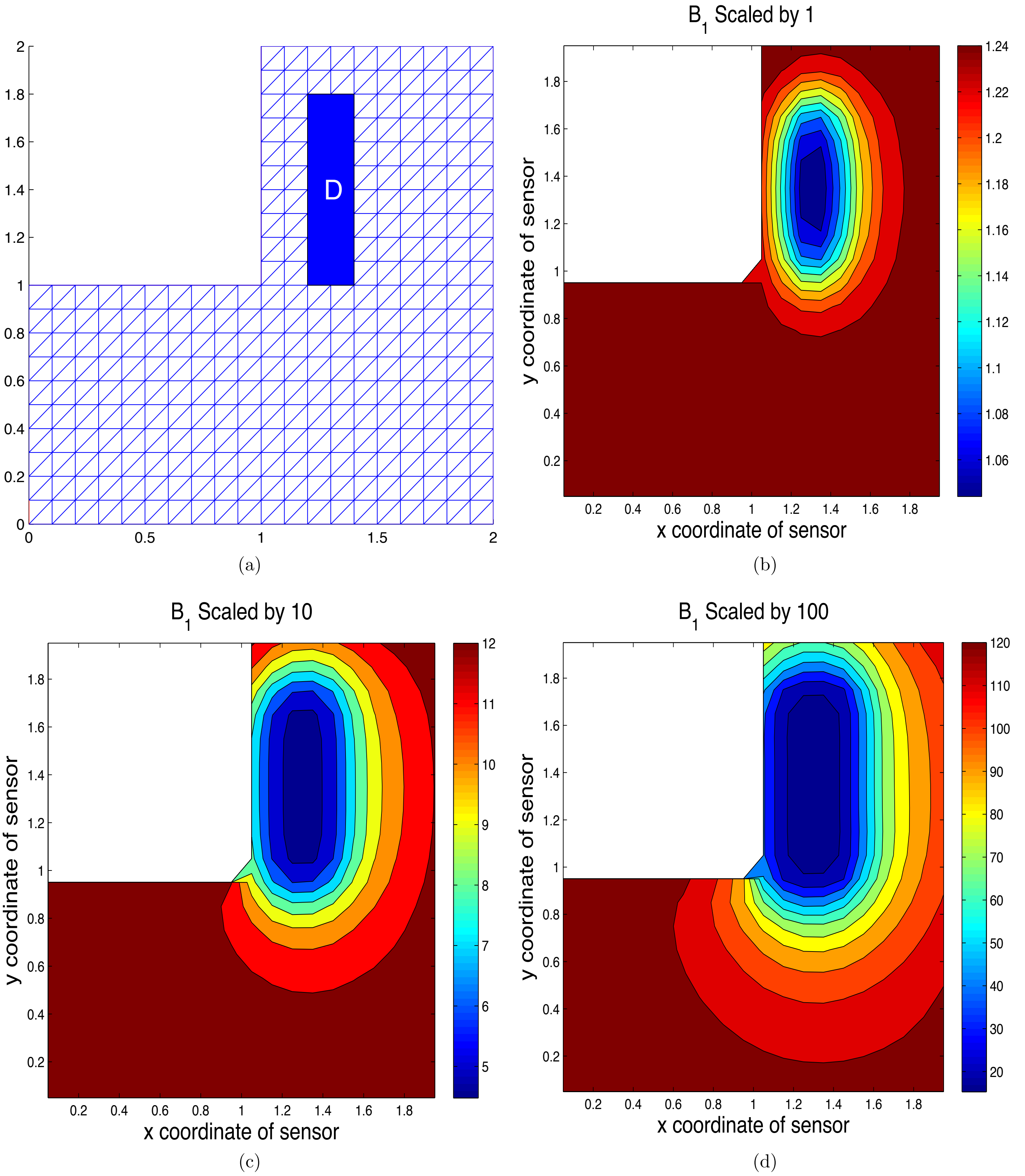
The zero Dirichlet boundary conditions guarantee that the matrix A is Hurwitz because the steady state solution is zero. If all the boundary conditions are changed to Neumann with zero flux, then one of the eigenvalues of A will be zero because the steady state solution changes to a constant that is not necessarily zero. The stabilizability of (A, B_2) needs to be verified as required by assumption A1 in section 3.1. However if at least one segment of the boundary has a constant negative flux or a Robin boundary condition where the there is a negative flux that depends on the state variable then the matrix A becomes Hurwitz again and assumption A1 in section 3.1 is satisfied.

The effect of scaling B_1, B_2 and σ are shown in Figures 5.21, 5.22 and 5.23. The effect of moving the actuator to a new location is shown in Figure 5.24. Scaling B_1 increases the difference between the cost at the optimal location and the cost in the area surrounding the optimal location. Scaling B_2 does not change the output feedback cost significantly. Decreasing σ expands the blue region. The optimal sensor location belongs to this region.

The discussion can be summarized as follows. The full control results in Figure 5.20 show that the optimal sensor location is at same place as the disturbance. This is in agreement with the analogous results for the optimal actuator location from [22] where it is found that the optimal actuator location is likely to be close to the disturbance location. In the output feedback results, the optimal sensor location is also at the disturbance location. Reducing the diffusivity parameter significantly increases the blue region and the optimal sensor location belongs to this area. This means for a smaller σ, the sensor can be placed further away from the disturbance and still an output feedback cost close to the optimal cost can be obtained.

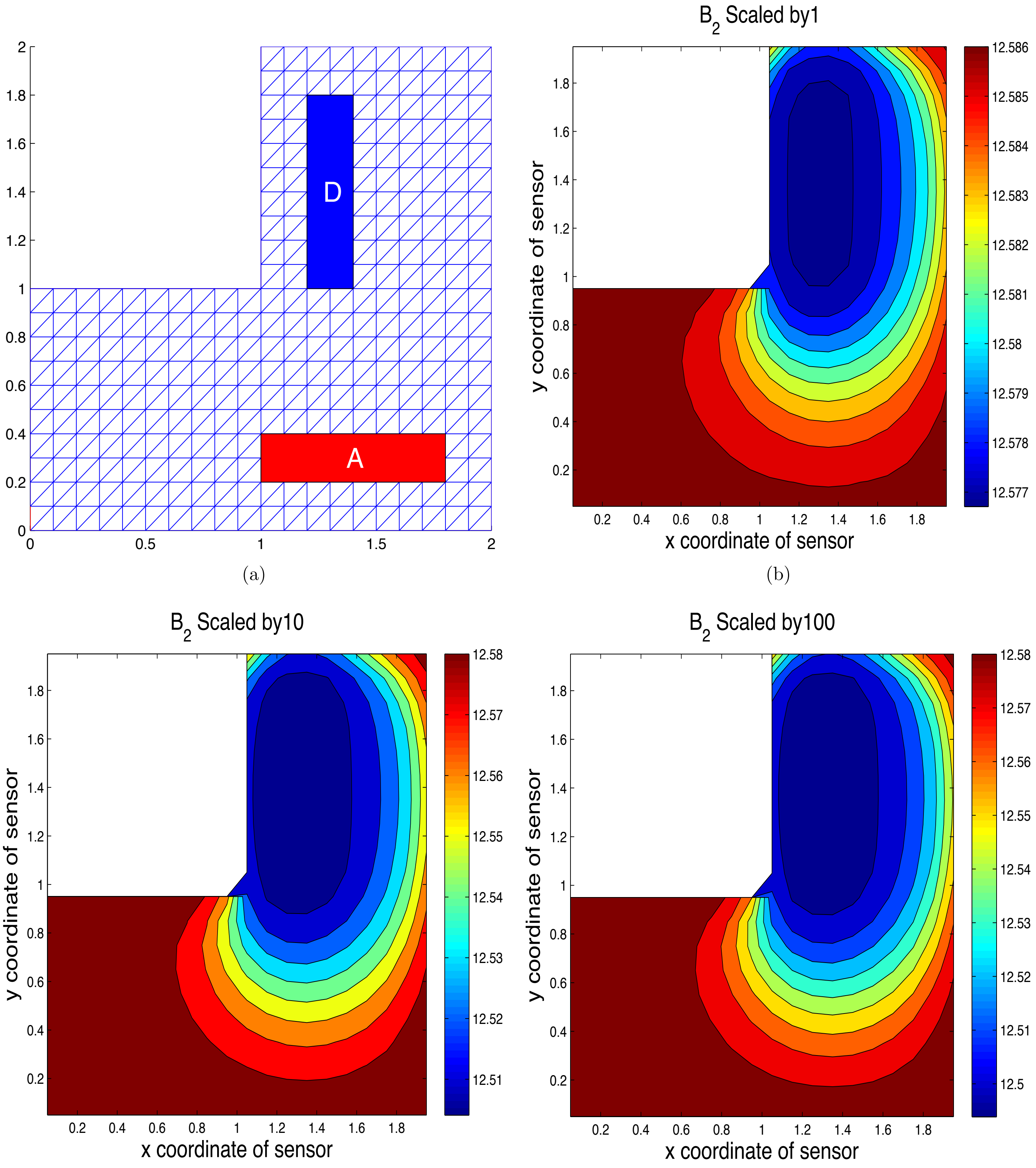
The L-shaped domain was chosen so that the domain is not complicated but also not as simple as a rectangle. The size of the disturbance and the actuator regions are modeling choices. The particular sizes chosen here are inspired by a window and a heater that cover part of a wall in a room. As the mesh is refined further, the computational costs grow. If there are multiple sensors on a high resolution mesh, the brute force method is computationally too intensive and an optimization method is needed. However in this work only one sensor is considered.

One issue is inverting the mass matrix, M, in (5.73). As the mesh is refined and M grows, matrix inversion becomes more time consuming. If the mesh is changed for each sensor position, the inverse of M needs to be calculated repeatedly. Furthermore, it is best to avoid matrix inversion in numerical methods because it is an unstable operation, however, typically M is sparse and it is possible to reorder the mesh vertices so that its nonzero elements are closer to the diagonal. In [28, chapter 3, page 160] a few algorithms are discussed that ensure M is block diagonal. Alternatively, it is possible to utilize Equation (5.72) directly which does not require the inverse of M. The MATLAB routine "care", for solving the algebraic Riccati equations (4.3) and (4.4) with A defined as in (5.74), has an option for doing so without explicitly inverting the mass matrix. Also , in [18], a technique for solving the H_∞ Riccati equations without inversion of the mass matrix is described.



(a) (b) (c) (d)

Figure 5.21: Figure(a) shows the disturbance B_1. In Figures(b-c-d) the full control cost is shown as a function of the sensor location. Matrix B_1 is scaled by $10^0, 10^1$ and 10^2 in the figures respectively. Diffusivity σ is 1 for all figures. Scaling B_1 expands the blue area. This area corresponds to the lowest full control cost. The optimal sensor location is inside the blue area.



(a)

(b)

(c)

(d)

Figure 5.22: Figure(a) shows the actuator B_2 and the disturbance B_1. Figures(b-c-d) show the output feedback cost as a function of the sensor position. Matrix B_2 is scaled by $10^0, 10^1$ and 10^2 in the figures respectively. Matrix B_1 is scaled by 10 and $\sigma = 1$ in all figures. The optimal sensor location is inside the blue region. Increasing the scaling factor for B_2 does not change the optimal sensor location significantly.

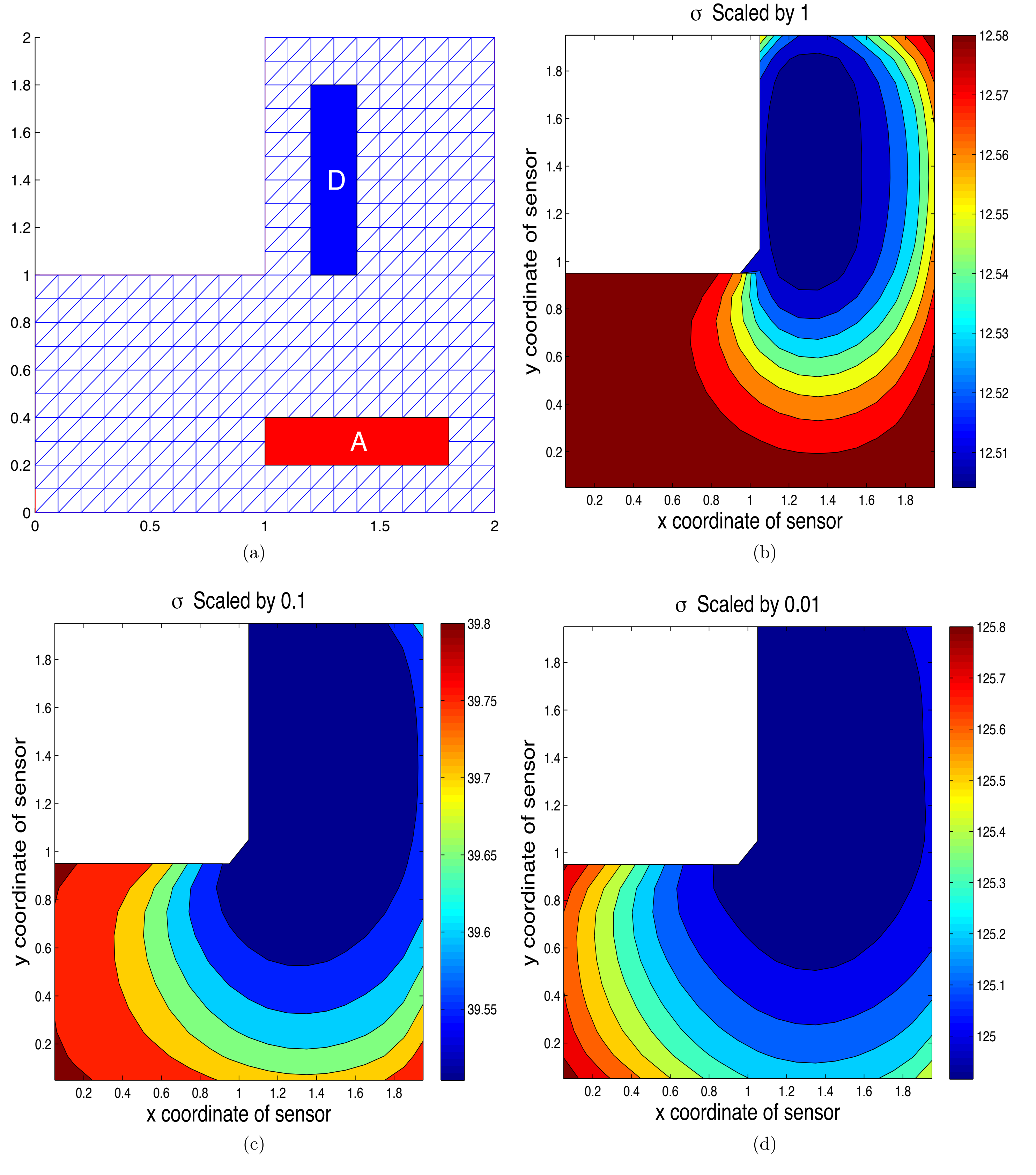
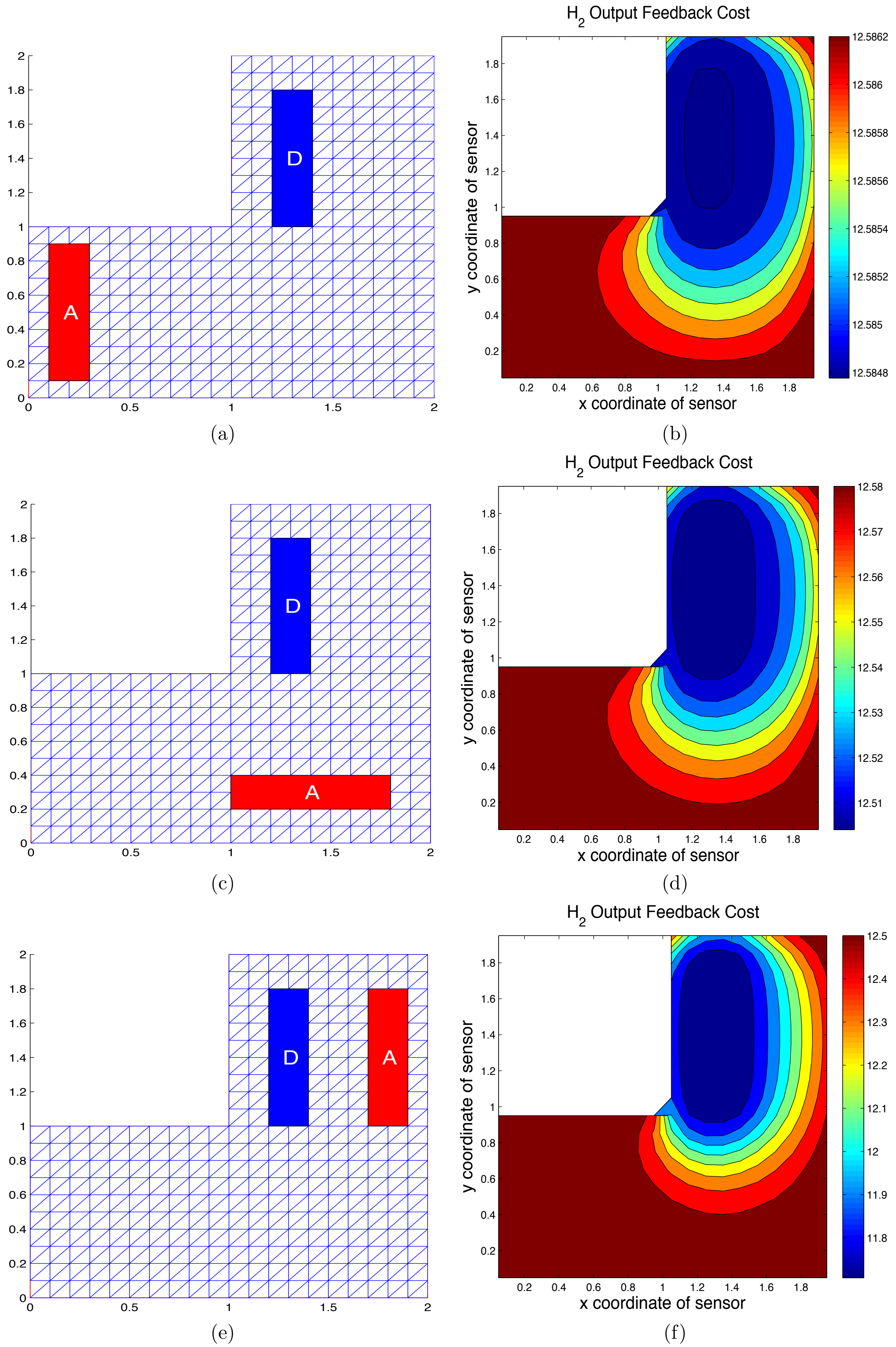


Figure 5.23: Figure(a) shows the disturbance B_1 and the actuator B_2. Figures(b-c-d) show the output feedback cost as a function of sensor location. In all figures B_1 and B_2 are scaled by 10. Parameter σ is $1, 0.1, 0.2$ in Figures(b-c-d) respectively. Decreasing σ expands the blue region. The optimal sensor location is inside the blue region.



(a) (b)

(c) (d)

(e) (f)

Figure 5.24: Figures(a-c-e) show the disturbance B_1 and the actuator B_2. Figures(b-d-f) show the corresponding output feedback costs. Moving the actuator to a different position does not change the costs significantly. The lowest cost in all graphs is in Figure(f) where the actuator is the closest to the disturbance.

Chapter 6

Conclusion and Future Work

In this work the H_2-optimal control strategy and the H_2-optimal sensor location problem are explained. The H_2-optimal control is designed for handling exogenous disturbing inputs that are white Gaussian. There are two cost functions associated with the H_2-optimal sensor location problem. These are the full control and the output feedback costs. In the full control cost only the performance of the sensor is considered while in the output feedback cost the combined performance of the sensor and the actuator is measured. The optimal sensor location is calculated for two systems. The first system is the one-dimensional Euler-Bernoulli beam equation with simply supported boundary conditions. Using the Fourier sine basis the state space equations are created. Different scenarios are considered with the disturbance and the actuator at various locations. The results show that the optimal sensor location tends to be close to the disturbance location. The actuator location affects the optimal sensor location in the direction of the actuator but not as strong as the disturbance. The second system is the two-dimensional diffusion equation in an L-shape region with zero Dirichlet boundary conditions. Using the finite element method the state space equations are created. Three different approaches for creating meshes are discussed and the structured mesh is chosen for the simulations. Because the problem is in two dimensions, the computational costs are higher compared to the beam system. The results show that the optimal sensor location is at the disturbance location. In contrast to the beam system, the actuator location does not affect the optimal sensor location significantly. The main computational difficulty is solving the Riccati equations involved in the H_2-costs repeatedly.

In this thesis the optimal location is found numerically, hence one possible extension is to find the optimal location using analytic techniques. Such a solution can potentially reduce the computational costs in a significant way. In this work only one sensor is considered. The computational costs increase if there are more sensors. The space of possible sensor positions grows exponentially with the number of sensors. An interesting extension would be to investigate the optimization algorithms for finding the optimal sensor location. One optimization algorithm for the optimal LQR actuator location can be found in [8].

Another interesting future project is to verify the results obtained here experimentally. Using a flexible beam or a system that obeys the diffusion equation one can experimentally test the different sensor locations and compare the findings to the ones predicted by theory. A signal generator would be needed to generate a white Gaussian signal.

Another control strategy that is designed for exogenous signals is H_∞. The control strategy in H_∞ is optimized to deal with the worst possible disturbance with respect to the input-output L_2-gain. The optimal sensor location problem can be reformulated for H_∞ and it would be interesting to compare the optimal location results for H_2 and H_∞.

From an application point of view, an interesting extension would be a 3-D convection-diffusion model with Robin boundary conditions. Such an equation can model a heater in a room. The ideal model would be a convective cell where the warm air circulates, however, such a model is nonlinear and would require nonlinear control theory.

Appendix - MATLAB Code

The MATLAB commands shown below create the L-shaped region and the mesh in Figure 5.13.

```
% first six are x coordinates, the next six are y coordinates
roomCornerCoordinates = [0,0,1,1,2,2,0,1,1,2,2,0];
% 2 is for polygon, 6 sides
room = [2,6,roomCornerCoordinates]’;
objects = [room];
[geometry,˜,˜,˜,˜] = decsg(objects); % encodes the geometry
[p,e,t] = initmesh(geometry); % creates the mesh
pdemesh(p,e,t); % plot the mesh
axis square; % makes the units on the axes equal length
```

The following MATLAB code creates a mesh such that the triangles fit the regions perfectly as in Figure 5.15.

```
% all the x coordinates come first, then the y coordinates
roomCornersCoord    = [0,0,1,1,2,2,0,1,1,2,2,0];
RegionSCornersCoord = [0.05,0.05,0.15,0.15,0.45,0.55,0.55,0.45];
RegionDCornersCoord = [1.1,1.1,1.2,1.2,1.05,1.95,1.95,1.05];
RegionACornersCoord = [1.05,1.05,1.95,1.95,0.1,0.2,0.2,0.1];

room = [2,6,roomCornersCoord]';% 2 is for polygon, 6 sides
S = [3,4,RegionSCornersCoord]';% 3 for rectangle, 4 sides
D = [3,4,RegionDCornersCoord]';
A = [3,4,RegionACornersCoord]';

% pad to enable concatenation, room is the longest
A = [A;zeros(length(room)-length(A),1)];
D = [D;zeros(length(room)-length(D),1)];
S = [S;zeros(length(room)-length(S),1)];
objects = [room,A,D,S];

geom = decsg(objects); % packing the geometric info
[p,e,t] = initmesh(geom);
pdemesh(p,e,t);
axis square;
```

The following MATLAB code shows how to refine the mesh around the region S as in Figure 5.16. The process for regions A and D is similar.

```
roomCornersCoord = [0,0,1,1,2,2,0,1,1,2,2,0];
room = [2,6,roomCornersCoord]’;
[geom,~,~,~,~] = decsg(room);
[p,e,t] = initmesh(geom);
triangles = FinaAllTrianglesThatNeedRefining(p,e,t);
[p,e,t]=refinemesh(geom,p,e,t,triangles,’regular’);
pdemesh(p,e,t);
--------------------------------------------
function triangles = FinaAllTrianglesThatNeedRefining(p,e,t)
%The syntax of polyxpoly needs the first number to repeated at the end again.
RegionS_XCornersCoord = [0.05,0.05,0.15,0.15,0.05];
RegionS_YCornersCoord = [0.45,0.55,0.55,0.45,0.45];
triangles = zeros(length(t),1);
counter = 1;
for i=1:length(t)
    Triangle_XCoord(1,1:4) = [p(1,t(1,i)),p(1,t(2,i)),p(1,t(3,i)),p(1,t(1,i))];
    Triangle_YCoord(1,1:4) = [p(2,t(1,i)),p(2,t(2,i)),p(2,t(3,i)),p(2,t(1,i))];
    %find the intersection between the triangle and the region
    IntersectWithS = polyxpoly(Triangle_XCoord,Triangle_YCoord,...
                               RegionS_XCornersCoord,RegionS_YCornersCoord);
    if(~isempty(IntersectWithS))
       triangles(counter) = i;
       counter = counter + 1;
    end
end
triangles(triangles==0) = []; % remove zeros;
end
```

References

[1] P. Ambrosio, F. Resta, and F. Ripamonti, "An H_2-norm approach for the actuator and sensor placement in vibration control of a smart structure," *Smart Materials and Structures*, vol. 21, no. 12, p. 125016, 2012.

[2] G. J. Balas and P. M. Young, "Sensor selection via closed-loop control objectives," *IEEE Transactions on Control Systems Technology*, pp. 692–705, 1999.

[3] O. A. Bauchau and J. I. Craig, *Structural analysis: with applications to aerospace structures.* Springer, 2009.

[4] I. Bruant and L. Proslier, "Optimal location of actuators and sensors in active vibration control," *Journal of Intelligent Material Systems and Structures*, pp. 197–206, 2005.

[5] V. Cerone, *Linear-quadratic control: an introduction.* Prentice Hall, 1995.

[6] K. K. Chen and C. W. Rowley, "H_2-optimal actuator and sensor placement in the linearised complex Ginzburg-Landau system," *Journal of Fluid Mechanics*, vol. 681, pp. 241–260, 2011.

[7] P. Colaneri, J. C. Geromel, and A. Locatelli, *Control theory and design: An RH_2 and RH_∞ Viewpoint.* Academic Press, 1997.

[8] N. Darivandi, K. A. Morris, and A. Khajepour, "An algorithm for LQ optimal actuator location," *Smart Materials and Structures*, vol. 22, no. 3, p. 035001, 2013.

[9] M. A. Demetriou and A. Armaou, "Optimal actuator placement and model reduction for a class of parabolic partial differential equations using spatial H_2-norm," in *Proceedings of the American Control Conference.* IEEE, 2005, pp. 4569–4574.

[10] K. H. Doki and G. Obinata, "Optimal sensor/actuator placement for active vibration control using explicit solution of algebraic Riccati equation," *Journal of Sound and Vibration*, pp. 1057–1075, 2000.

[11] J. C. Doyle, K. Glover, P. P. Khargonekar, and B. A. Francis, "State-space solutions to standard H_2 and H_∞ control problems," *IEEE Transactions on Automatic Control*, vol. 34, no. 8, pp. 831–847, 1989.

[12] J. E. Flaherty, *Course notes - finite element analysis.* http://www.cs.rpi.edu/~flaherje/pdf/fea1.pdf, 2005.

[13] B. Frances, "A course in H_∞ control theory," *Lecture Notes in Control and Information Sciences*, vol. 88, 1987.

[14] B. Friedland, *Control system design: an introduction to state-space methods.* Courier Dover Publications, 2012.

[15] T. W. Gamelin, *Complex analysis.* Springer, 2001.

[16] A. Ha and L. Liu, "Sensor and actuator location in motion control of flexible structures," *Journal of Sound and Vibration*, pp. 239 – 261, 1993.

[17] T. J. Hughes, *The finite element method: linear static and dynamic finite element analysis.* Courier Dover Publications, 2012.

[18] D. Kasinathan, K. Morris, and S. Yang, "Solution of large generalized H_∞ algebraic Riccati equations," *Journal of Computational Science*, vol. 5, no. 3, pp. 517–526, 2014.

[19] S. Kondoh, C. Yatomi, and K. Inoue, "The positioning of sensors and actuators in the vibration control of flexible systems," *JSME International Journal, Ser, 3, Vibration, Control Engineering, Engineering for Industry*, 1990.

[20] C. Kubrusly and H. Malebranche, "Sensors and controllers location in distributed systems-a survey," *Automatica*, 1985.

[21] D. Liberzon, *Calculus of variations and optimal control theory: a concise introduction.* Princeton University Press, 2012.

[22] K. A. Morris, M. Demetriou, and S. Young, "Using H_2-control performance metrics for the optimal actuator location of distributed parameter systems," *IEEE Transactions on Automatic Control.*

[23] K. A. Morris, *Introduction to feedback control.* Academic Press, Inc., 2000.

[24] K. A. Morris, "Control of systems governed by partial differential equations," *The Control Theory Handbook*, 2010.

[25] K. A. Morris, "Linear-quadratic optimal actuator location," *IEEE Transactions on Automatic Control*, vol. 56, no. 1, pp. 113–124, Jan 2011.

[26] R. Muradore, F. Bezzo, and M. Barolo, "Optimal sensor location for distributed-sensor systems using multivariate regression," *Computers & chemical engineering*, pp. 521–534, 2006.

[27] C. Pozrikidis, *Introduction to finite and spectral element methods using MATLAB.* CRC Press, 2005.

[28] H. R. Schwarz, J. Whiteman, and C. M. Whiteman, *Finite element methods.* Academic Press London, 1988.

[29] N. D. Shoushtari, "Optimal active control of flexible structures, applying piezoelectric actuators," Ph.D. dissertation, University of Waterloo, 2013.

[30] R. C. Smith, *Smart material systems: model development.* Siam, 2005.

[31] K. Zhou, J. Doyle, and K. Glover, *Robust and optimal control.* Prentice Hall, 1996.