Optimal Linear Feedback Control for Incompressible Fluid Flow

Miroslav K. Stoyanov

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science
in
Mathematics.

Jeff Borggaard, Chair
Lizette Zietsman
John Burns

May 24, 2006
Blacksburg, Virginia

Optimal Linear Feedback Control for Incompressible Fluid Flow

Miroslav K. Stoyanov

(ABSTRACT)

Linear feedback control is considered for large systems of differential algebraic equations arising from discretization of saddle point problems. Necessary conditions are derived by applying the Maximum Principle and have the form of constrained Riccati equations. We consider two approaches for solving the feedback control problem as well as practical numerical methods. Numerical studies using examples derived from a constrained heat equation and Stokes equation confirms the effectiveness of the approaches we consider.

To my wife, for her infinite support.

Acknowledgments

I wish to express my thanks to my advisor Dr. Borggaard, for his help and guidance in the past two years. To Dr. Zietsman for helping me understanding Riccati solvers. To Dr. Burns for giving me real appreciation of control theory.

Thanks to Dr. Adjerid and Dr. Iliescu for teaching me numerical analysis and answering my numerous random questions.

I wish to thank my parents for always encouraging me in my studies in math.

Last but not least, I wish to thank my wife for her infinite support.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Feedback Control for DAEs

A typical linear control problem involves a linear system of ordinary differential equations (ODE) of the form

$$\dot{x}(t) = Ax(t) + Bu(t), \quad x(0) = x_0. \tag{1.1}$$

The main objective is to find $u(\cdot)$ in some admissible set that will minimize a given functional cost. The standard Linear-Quadratic case involves no explicit restriction on neither $u(\cdot)$ nor the final conditions for $x(\cdot)$, but rather a quadratic cost functional of the form

$$J(u(\cdot)) = \int_0^T \langle x(t), Qx(t)\rangle + \langle u(t), Ru(t)\rangle \, dt,$$

where Q is symmetric positive semi-definite and R is symmetric positive definite. In the Linear-Quadratic problem, the objective is to find an optimal $u(\cdot)$ that minimizes $J(\cdot)$ subject to the constraint (1.1). In the case where the final time is infinite ($T = \infty$), the problem is called the Linear-Quadratic-Regulator (LQR) problem.

The problem given above is well studied [11, 5, 10] and the solution is a feedback control of the form $u(t) = -K(t)x(t)$. The optimal gain $K(t)$ is given by $K(t) = R^{-1}B^T\Pi(t)$, where $\Pi(t)$ is the solution to either the Riccati Differential Equation (RDE), if $T < \infty$,

$$-\dot{\Pi}(t) = A^T\Pi(t) + \Pi(t)A - \Pi(t)BR^{-1}B^T\Pi(t) + Q, \quad \Pi(T) = 0,$$

or the Riccati Algebraic Equation (RAE), if $T = \infty$,

$$0 = A^T\Pi + \Pi A - \Pi BR^{-1}B^T\Pi + Q.$$

Methods for solving these equations have been studied and different numerical techniques devised [1, 5]. The main difficulty in solving these equations is their stability and amount of computational work involved.

A common approach in finding an optimal control for systems of partial differential equations (PDE), is to use some linearization of the PDE and then finite difference or finite element discretization of the spatial domain. The result is a system of linear ODEs and the control for that system can be found with the techniques mentioned above. Under a reasonable set of conditions the control for the ODE system will generate approximations of the control for the PDE system.

The main focus of this thesis is another class of control problems. In the discretization of PDE problems we often have equations of the form

$$E\dot{x}(t) = Ax(t) + Bu(t). \tag{1.2}$$

When E is invertible, the problem is equivalent to a problem of the form (1.1). The case when E is singular gives rise to a coupled system of differential and algebraic equations (DAE), also known as singular systems, descriptor systems and semi-state systems among others [4]. The DAE system is not equivalent to (1.1) and an alternative approach has to be taken.

The DAE system that we consider, has the form

$$\begin{pmatrix} E & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{pmatrix} = \begin{pmatrix} A_{11} & A_{21}^T \\ A_{21} & 0 \end{pmatrix} \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} + \begin{pmatrix} B \\ 0 \end{pmatrix} u(t), \tag{1.3}$$

where E is symmetric positive definite. DAE systems with such structure, come from discretization of saddle point problems. One example is the finite element approximation of Stokes and Oseen equations, another example comes from solving a specific discretization of the heat equation.

1.2 Two Examples

1.2.1 Discretizations of Incompressible Flow Problems

The Stokes and Oseen equations are linearizations of the Navier-Stokes equations that model incompressible fluid flow. The Navier-Stokes equations are

$$\begin{aligned} \vec{u}_t &= \frac{1}{Re}\Delta\vec{u} - \langle \vec{u}, \nabla \rangle \vec{u} - \nabla p \\ 0 &= \nabla \cdot \vec{u}. \end{aligned}$$

The Stokes equations linearize Navier-Stokes around 0 and the Oseen around prescribed incompressible flow U. Thus the Stokes equations are

$$\begin{aligned}\vec{u}_t &= \frac{1}{Re}\Delta\vec{u} - \nabla p \\ 0 &= \nabla\cdot\vec{u},\end{aligned}$$

and the Oseen equations are

$$\begin{aligned}\vec{u}_t &= \frac{1}{Re}\Delta\vec{u} - \langle U, \nabla\rangle\,\vec{u} - \langle\vec{u}, \nabla\rangle\,U - \nabla p \\ 0 &= \nabla\cdot\vec{u}.\end{aligned}$$

In all four cases $\vec{u}$ is a vector function for the velocity of the fluid in different directions and p is the pressure. The second equation in the system above is independent of time and corresponds to the algebraic term in the DAE. The divergence term does not depend explicitly on p and thus the lower right block of the DAE matrix is 0. The operator gradient acting on p and the divergence operator acting in the constrained equation are adjoint, thus we have the relationship between the two blocks in the DAE system, $A_{12} = A_{21}^T$. In addition, for the Stokes problem we have the weak form of the Laplacian operator acting on u. Since the weak form is self-adjoint the matrix A_{11} will be symmetric using Galerkin finite elements.

1.2.2 Heat Equation with Imposed BC Constraints

The heat equation is given by

$$u_t = \Delta u.$$

If we wish to impose 0 boundary conditions, we can add an algebraic part to the equation of the form

$$u|_{\partial\Omega} = 0,$$

where Ω is the specified domain. The second equation does not depend explicitly on the time and thus it corresponds to the algebraic part of the DAE system. In terms of finite elements, the above equation can be viewed as minimizing over a set of test functions that do not vanish on the boundary of the domain. Then the structure of the discrete equation will be consistent with (1.3).

In both discretizations the matrix E is a symmetric positive definite mass matrix. For the rest of the thesis, we shall assume that E has such form, however, many of the results can extend to more general settings.

1.3 Literature Survey

The general structure of the DAE Linear-Quadratic control problem is discussed in many works including [2] and [6]. In [2], necessary conditions for the optimal control are derived and the corresponding Riccati equations given. The approach taken by [2] is to transform the general DAE system (1.2) into a form similar to (1.3), then an assumption for the lower right block of A is made. The assumption is that the block is invertible. Under that assumption, equations for the control could be derived. If the assumption fails, in general, the Riccati Differential Equation could have jump discontinuities (impulses) and thus make it impossible to solve via any reasonable numerical method.

The DAE system (1.3) fails the invertibility assumption, thus we cannot directly apply the results from [2, 6]. In this thesis we substitute the invertibility assumption with the assumption that the lower right part of the block is 0, $A_{12} = A_{21}^T$ and E is symmetric positive definite. In that case we can show that there are no jump discontinuities in the Riccati equation and feedback control can be derived.

The Riccati equation that gives the solution to (1.3) is different from the standard Riccati equation. The structure of the standard Riccati equations has been studied, and stable and efficient methods for solving (RAE) have been devised [1, 5, 9]. Since the properties of the new Riccati Equation are unknown, we would wish to convert it to or approximate it by a standard Riccati Equation.

1.4 Our Approach

In this section, we describe our approach for solving the feedback control problem. A popular approach for simulating these problems is to impose the algebraic constraints using a penalty method approach [8]. Thus, instead of equations of the form (1.3), we consider

$$\begin{pmatrix} E & 0 \\ 0 & 0 \end{pmatrix} \overset{\cdot}{\begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix}} = \begin{pmatrix} A_{11} & A_{21}^T \\ A_{21} & \epsilon M \end{pmatrix} \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} + \begin{pmatrix} B \\ 0 \end{pmatrix} u(t) \qquad (1.4)$$

for small values of ϵ. The matrix M should be easily invertible (e.g. the identity matrix, a sparse mass matrix, etc.). With this approximate system, we now consider the control problem for

$$E\dot{x}_1(t) = \left(A_{11} - \frac{1}{\epsilon}A_{21}^T M^{-1} A_{21}\right) x_1(t) + Bu(t), \tag{1.5}$$

and seek $u_\epsilon(t) = -K_\epsilon(t)x_1(t)$.

There are many natural questions that arise:

- Does $u_\epsilon \to u$ and if it does can we say that $K_\epsilon \to K$?
- Since the resulting system is still large, can we develop an efficient algorithm that takes advantage of problem structure (such as sparsity) and modern computer architectures (such as parallel computer clusters).

In Chapter 3, we answer the first question in the affirmative. This is also seen in several numerical experiments in Chapter 5.

The discretization of a PDE can result in a very large sparse system of equations. We wish to look for a numerical method that solves standard Riccati equations, takes advantage of sparsity and can be efficiently implemented for a parallel architecture. This is an area with a large body of current research [Reference]. The development of two efficient methods is discussed in Chapter 4, where we discuss approaches based on Chandrasekhar equations and the matrix sign function.

1.5 Thesis Overview

The rest of the thesis is organized as follows. Chapter 2 provides preliminary results that follow from the structure of the DAE problem. Chapter 3 gives a detailed derivation of the necessary condition for the DAE linear feedback control problem and alternative ways to approximate the solution. Chapter 4 discusses ways to solve the standard Riccati equation for large systems using sparse operations or parallel architecture. Chapter 5 provides numerical results and finally we provide some conclusions in Chapter 6.

Chapter 2

Properties of the DAE system

In this chapter we consider a general differential algebraic equation (DAE) system of the form

$$\begin{pmatrix} E & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{pmatrix} = \begin{pmatrix} A_{11} & A_{21}^T \\ A_{21} & 0 \end{pmatrix} \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix}, \tag{2.1}$$

where E is symmetric positive definite and $dim\,(ker(A_{21})) > 0$. We wish to address questions about the existence and uniqueness of the solution as well as basic linear algebra results involving the $A_{12} = A_{21}^T$ structure.

2.1 General Linear Algebra Results

In order to consider a control problem on any system of equations, we first need a well posed system of equations. In order for (2.1) to be well posed, it is necessary for A_{21}^T to have full column rank.

Lemma 1 ***Rank***

If system (2.1) is well posed, then A_{21}^T has full column rank.

Proof: *Let $\hat{x}_1(\cdot)$, $\hat{x}_2(\cdot)$ be the unique solution to (2.1). Assume to the contrary that $A_{21}^T\hat{p} = 0$ and $\hat{p} \neq 0$. Let $f : \boldsymbol{R} \rightarrow \boldsymbol{R}$ be any scalar function with $f(0) = 0$ and $f \not\equiv 0$. Then given some initial conditions $x_1(0) = x_1^0$, (2.1) will have unique solution $\hat{x}_1(\cdot)$ and $\hat{x}_2(\cdot)$. However, $\tilde{x}_1(t) = \hat{x}_1(t)$ and $\tilde{x}_2(t) = \hat{x}_2(t) + f(t)\hat{p}$ satisfies the same initial conditions and also satisfies (2.1), this contradicts the well posedness assumption. Therefore, if $A_{21}^T\hat{p} = 0$, then $\hat{p} = 0$.*

Since it is a necessary condition for well posedness, for the rest of the thesis we can assume A_{21}^T that has full column rank.

Given that E is symmetric positive definite, we observe the following result.

Lemma 2 ***Invertibility***

If S has full column rank and if E is definite, then

$S^T ES$ is invertible.

Proof: *Suppose $S^T ESx = 0$ for some vector x. Then*

$$0 = \left\langle S^T ESx, x \right\rangle = \langle ESx, Sx \rangle .$$

By the definite property of E, the above implies that $Sx = 0$. Since S has full column rank, that implies that $x = 0$. Therefore, if $S^T ESx = 0$, then $x = 0$.Since the kernel of $S^T ES$ is trivial, $S^T ES$ is invertible.

The algebraic part of system (2.1) states that at any time t, $x_1(t) \in ker(A_{21})$. We want to explore the properties of the kernel of A_{21}.

Lemma 3 ***Orthogonality Lemma***

$$ker(A_{21}) \perp range(A_{21}^T)$$

Proof: *Let $c \in ker(A_{21})$ and $b \in range(A_{21}^T)$ i.e. $b = A_{21}^T x$ for some x and $A_{21}c = 0$. Then $\langle c, b \rangle = \left\langle c, A_{21}^T x \right\rangle = \langle A_{21}c, x \rangle = \langle 0, c \rangle = 0$. Thus $ker(A_{21}) \perp range(A_{21}^T)$.*

The above lemma can be extended to the following.

Lemma 4 ***Separation of the Null Space***

If $x \perp range(A_{21}^T)$ then $x \in ker(A_{21})$.

If $x \perp ker(A_{21})$ then $x \in range(A_{21}^T)$.

Proof: *If $x \perp range(A_{21}^T)$, then $x^T A_{21}^T = 0$, therefore, $A_{21}x = 0$. Thus we have established the first proposition.*

For the second part observe, that $x = 0$ is orthogonal to any subspace and is in any subspace, so the proposition holds trivially for $x = 0$. Now suppose

$x \perp ker(A_{21})$, $x \neq 0$ and $x \notin range(A_{21}^T)$. Then x can be split into $x = x_r + x_p$ where $x_r \in range(A_{21}^T)$ and $x_p \perp range(A_{21}^T)$. By the first part of this Lemma $x_p \in ker(A_{21})$, therefore since $x \perp ker(A_{21})$, $x_p = 0$. Thus $x = x_r \in range(A_{21}^T)$. Which gives a contradiction. Thus if $x \perp ker(A_{21})$, then $x \in range(A_{21}^T)$.

The kernel of A_{21} plays important role in our analysis. In many places we wish to form a matrix V such that the columns of V form a minimal orthonormal basis for $ker(A_{21})$. In practice we can form V using SVD singular value decomposition) or QR decomposition. We also need to form $\bar{V}$ so that $\bar{V}$ form a basis for $ker(A_{21})^{\perp}$ (i.e. $\bar{V}$ is orthogonal to V). Next we observe some of the properties of such basis V.

Lemma 5 ***Invertibility Within the Kernel***

$$\textit{If } a, b \in ker(A_{21}) \textit{ and } V^T a = V^T b,$$

$$\textit{then } a = b$$

Proof: *Since $a, b \in ker(A_{21})$, $a = Va_v$ and $b = Vb_v$. Then $V^T a = V^T V a_v = a_v$ and $V^T b = V^T V b = B_v$. Therefore, $a_v = b_v$, which implies $a = b$.*

Lemma 6 ***Kernel Identity***

$$\textit{If } x \in ker(A_{21}) \textit{ then } VV^T x = x$$

Proof: *Let $x \in ker(A_{21})$, then let $b = VV^T x$ and by the properties of V, $b \in ker(A_{21})$. Then multiply both sides by V^T, follows that $V^T x = V^T b$. Since both $x, b \in ker(A_{21})$ by the Kernel Invertibility Lemma 5, $x = b$.*

2.2 DAE Properties

For a general linear system of differential algebraic equations we have the following theorem [4]

Theorem 1 ***Well Posed General DAE***

Consider the DAE system of the form

$$E\dot{x}(t) = Ax(t) + f(t),$$

where E is singular. The system if well posed if and only if all of the following three conditions hold *i) det$(sE - A)$ is not uniformly 0 for all $s \in \boldsymbol{R}$ (we can take Laplace transforms)*

ii) the initial conditions are consistent with the algebraic constraint

iii) $f(t)$ is differentiable.

For the purposes of the control problem we will assume that the initial conditions are always consistent: $A_{21}x_1^0 = 0$. Furthermore, in general we assume that there is no forcing function (i.e. $f(t) \equiv 0$). In some cases, the control acts as a forcing function, but we will show that the optimal control is a feedback control and thus it is differentiable.

We wish to use Theorem 1 to prove that any DAE system of the form (2.1), where E is symmetric positive definite and A_{21}^T has full column rank is well posed.

Lemma 7 ***Well Posed DAE***

If a DAE system has the form (2.1), with E being symmetric positive definite, A_{21}^T having full column rank and consistent initial conditions, then it is well posed.

***Proof**: If det$(sE - A) \equiv 0$ for all $s \in \boldsymbol{R}$, then for every s, there is a vector $x(s) = \left(x_1^T, x_2^T\right)$, so that $x(s)$ is an eigenvector of $sE - A$ corresponding to the eigenvalue 0.*

$$\begin{pmatrix} sE - A_{11} & -A_{21}^T \\ -A_{21} & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = 0.$$

If $x_1 = 0$, then $-A_{21}^T x_2 = 0$, therefore by the full rank of A_{21}^T, $x_2 = 0$, therefore $\|x\| = 0$, therefore x is not an eigenvector. Thus, we take $x_1 \neq 0$. From $A_{21}x_1 = 0$, we can conclude that $x_1 \in ker(A_{21})$. Since

$$(sE - A_{11})\, x_1 - A_{21}^T x_2 = 0,$$

we have that

$$x_1^T (sE - A_{11})\, x_1 - x_1^T A_{21}^T x_2 = 0.$$

By Lemma 3, $x_1^T A_{21}^T = 0$, therefore

$$s\left(x_1^T E x_1\right) - x_1^T A_{11} x_1 = 0.$$

Without loss of generality, we can assume that $\|x_1\| = 1$, therefore

$$s\left(x_1^T E x_1\right) - x_1^T A_{11} x_1 > s\frac{1}{\lambda} - \|A_{11}\|,$$

where λ is the smallest eigenvalue of E. This bound does not depend on x_1 and since both A_{11} and E are constant matrixes, we can pick $\boldsymbol{s}$ *big enough so that*

$$s\left(x_1^T E x_1\right) - x_1^T A_{11} x_1 > 0$$

for all $x_1 \in ker(A_{21})$. Therefore we satisfy the condition in Theorem 1 and the DAE is well posed.

Chapter 3

Necessary Condition for DAE

Consider a system of Differential Algebraic Equations (DAE) of the form:

$$\begin{pmatrix} E & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{pmatrix} = \begin{pmatrix} A_{11} & A_{21}^T \\ A_{21} & 0 \end{pmatrix} \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} + \begin{pmatrix} B \\ 0 \end{pmatrix} u(t) \tag{3.1}$$

$$x_1(0) = x_1^0 \qquad \text{and} \qquad x_2(0) = x_2^0. \tag{3.2}$$

This system can also be written in the form

$$\begin{aligned} E\dot{x_1}(t) &= A_{11}x_1(t) + A_{21}^T x_2(t) + Bu(t) \\ 0 &= A_{21}x_1(t) \end{aligned}$$

where E is symmetric, positive definite and A_{21}^T has full column rank.

We wish to find the control $u(\cdot)$ that will minimize the quadratic cost functional, with properties given in Chapter 1,

$$J(u(\cdot)) = \frac{1}{2}\int_0^\infty \langle x_1(t), Qx_1(t)\rangle + \langle u(t), Ru(t)\rangle \, dt. \tag{3.3}$$

It will be shown that the optimal control $u(\cdot)$ is linear feedback of the form

$$u(t) = -Kx_1(t). \tag{3.4}$$

The feedback matrix K can be determined as $K = R^{-1}B^T\Pi$ where Π is the solution to a Riccati equation. In this chapter we will discuss three ways of finding the optimal gain K or, in particular, Π. The first way is by directly applying the Maximum Principle. The second is to perturb the original system and convert it to a purely differential system. The third is to do a change of variable and thus eliminate the algebraic part.

3.1 Direct Application of the Maximum Principle

First we consider the case with finite time. Here we have the functional

$$J(x_1(\cdot), x_2(\cdot), u(\cdot)) = \frac{1}{2}\int_0^T \langle x_1(t), Qx_1(t)\rangle + \langle u(t), Ru(t)\rangle \, dt$$

that we wish to minimize subject to

$$Ex_1(t) - x_1^0 - \int_0^t A_{11}x_1(\tau) + A_{21}^T x_2(\tau) + Bu(\tau)d\tau = 0$$

$$A_{21}x_1(t) = 0.$$

We can write the above as an optimization problem over a Banach space. Let n corresponds to the dimension of the vector x_1, l corresponds to the dimension of the vector x_2 and m corresponds to the dimension of the control u, then let

$$X_1 = \{x : \mathbf{R} \to \mathbf{R}^n \colon x_i(\cdot) \in \mathbf{C}([0,T])\}$$

$$X_2 = \left\{x : \mathbf{R} \to \mathbf{R}^l \colon x_i(\cdot) \in \mathbf{L}^1([0,T])\right\}$$

$$U = \left\{u : \mathbf{R} \to \mathbf{R}^m \colon u_i(\cdot) \in \mathbf{L}^1([0,T])\right\}$$

$$Z_2 = \left\{z : \mathbf{R} \to \mathbf{R}^l \colon z_i(\cdot) \in \mathbf{C}([0,T])\right\}.$$

Therefore, the functional $J(\cdot)$ acts on

$$J : X_1 \times X_2 \times U \to \mathbf{R}.$$

Define the constraint function $H(\cdot)$ to be

$$H(x_1(\cdot), x_2(\cdot), u(\cdot)) = \begin{pmatrix} Ex_1(t) - x_1^0 - \int_0^t A_{11}x_1(\tau) + A_{21}^T x_2(\tau) + Bu(\tau)d\tau \\ A_{21}x_1(t) \end{pmatrix}$$

Therefore, the constraint acts on

$$H : X_1 \times X_2 \times U \to X_1 \times Z_2.$$

Using the above notation, we can write the minimization problem as

$$\text{minimize } J\left(x_1(\cdot), x_2(\cdot), u(\cdot)\right)$$

$$\text{subject to } H\left(x_1(\cdot), x_2(\cdot), u(\cdot)\right) = 0.$$

Assuming that the minimum exists and that $x_1^*(\cdot)$, $x_2^*(\cdot)$ and $u^*(\cdot)$ are optimal, then according to the Lagrange Multiplier Theorem [11], there exist a constant λ_0^* and a bounded linear functional λ^* acting on $\lambda^* : X_1 \times Z_2 \to \mathbf{R}$ so that the Lagrangian function

$$L\left(x_1(\cdot), x_2(\cdot), u(\cdot)\right) = \lambda_0^* J\left(x_1(\cdot), x_2(\cdot), u(\cdot)\right) + \lambda^* H\left(x_1(\cdot), x_2(\cdot), u(\cdot)\right)$$

has a stationary point at $x_1^*(\cdot)$, $x_2^*(\cdot)$ and $u^*(\cdot)$. Furthermore, λ_0^* and λ^* are not simultaneously zero. According to the Riesz Representation Theorem [12], a bounded linear functional on the space of continuous functions over a compact set can be represented as an integral with respect to unique Baire measure. Baire measures on compact sets are generated by distribution functions that are of bounded variation and continuous to the right. Thus, given functional λ^*, there exists vector functions λ_1^* and λ_2^* so that

$$\lambda^*\left(x_1(\cdot), z_2(\cdot)\right) = \int_0^T d\lambda_1^{T*} x_1(t) + \int_0^T d\lambda_2^{T*} z_2(t).$$

Furthermore, we can chose λ_1^* and λ_2^* so that $\lambda_1^*(T) = 0$ and $\lambda_2^*(T) = 0$, because the distribution functions of Baire measures are unique up to a constant [11, 12].

Using this notation we can rewrite the Lagrangian function as

$$L\left(x_1(\cdot), x_2(\cdot), u(\cdot)\right) = \frac{1}{2}\int_0^T \lambda_0^* \left\langle x_1(t), Qx_1(t)\right\rangle + \lambda_0^* \left\langle u(t), Ru(t)\right\rangle dt$$

$$+ \int_0^T d\lambda_1^{T*} \left(Ex_1(t) - x_1^0 - \int_0^t A_{11}x_1(\tau) + A_{21}^T x_2(\tau) + Bu(\tau) d\tau \right)$$

$$+ \int_0^T d\lambda_2^{T*} A_{21} x_1(t).$$

The Lagrange Multiplier Theorem states that the Lagrangian function will have a stationary point at the optimal control $u^*(\cdot)$ and the optimal trajectory $x_1^*(\cdot)$and $x_2^*(\cdot)$. In other words, the variation of L in any arbitrary direction

$$(h(\cdot), p(\cdot), s(\cdot)) \in X_1 \times X_2 \times U$$

is zero. Note that in order to be consistent with initial conditions for $x_1(\cdot)$, we need to take variations in the direction of $h(\cdot)$ so that $h(0) = 0$.

Variations for $x_1(\cdot)$ and $x_2(\cdot)$ give us

$$\int_0^T \lambda_0^* x_1^{T*}(t) Qh(t) dt + \int_0^T d\lambda_1^{T*} \left(Eh(t) - \int_0^t A_{11}h(\tau) d\tau \right) + \int_0^T d\lambda_2^{T*} A_{21} h(t) = 0,$$

$$\int_0^T d\lambda_1^{T*} \left(\int_0^t A_{21}^T p(\tau) d\tau \right) = 0.$$

The above has to be true for all functions $h(\cdot) \in X_1$ with $h(0) = 0$ and thus it will be true for all piecewise smooth functions with $h(0) = 0$. In this case, we can integrate by parts in both expressions and obtain

$$\int_0^T \lambda_0^* x_1^{T*}(t) Q h(t) dt - \int_0^T \lambda_1^{T*}(t) \left(E\dot{h}(t) - A_{11} h(t) \right) dt - \int_0^T \lambda_2^{T*}(t) A_{21} \dot{h}(t) dt = 0$$

$$\int_0^T \lambda_1^{T*}(t) A_{21}^T p(t) dt = 0.$$

The second expression is true for all $p(\cdot) \in X_2$ and therefore by the Fundamental Lemma of Calculus of Variation (FLCV) [11]

$$A_{21} \lambda_1^*(t) = 0 \tag{3.5}$$

almost everywhere in $[0, T]$. Furthermore, by right continuity of $\lambda_1^*(\cdot)$ it is true everywhere. Also by the FLCV the first expression is equivalent to

$$-\left(E\lambda_1^*(t) + A_{21}^T \lambda_2^*(t) \right) = \int_0^t A_{11}^T \lambda_1^*(\tau) + \lambda_0^* Q x_1^*(\tau) d\tau. \tag{3.6}$$

We wish to differentiate both sides of (3.6) and obtain a differential equation for $\lambda_1^*(\cdot)$ and $\lambda_2^*(\cdot)$. We know that $\lambda_1^*(\cdot)$ and $\lambda_2^*(\cdot)$ are functions of bounded variation and thus differentiable almost everywhere, we can also conclude that

$$-E\dot{\lambda}_1^*(t) = A_{11}^T \lambda_1^*(t) + \lambda_0^* Q x_1^*(t) + A_{21}^T \dot{\lambda}_2^*(t),$$

however, in order for us to have a differential equation, we also need to know that

$$\lambda_1^*(t) = \int_0^t \dot{\lambda}_1^*(\tau) d\tau.$$

The above will follow if we know that $\lambda_1^*(\cdot)$ is an absolutely continuous function. To establish that result we need to go through several steps (for more of the properties of absolutely continuous functions and functions of bounded variations see [12]).

We can observe that on the right hand side of (3.6) we do indeed have an absolutely continues function. Any linear combination of absolutely continuous function is also absolutely continuous. Since E is non-singular we can multiply (3.6) by E^{-1} and obtain

$$-\left(\lambda_1^*(t) + E^{-1} A_{21}^T \lambda_2^*(t) \right) = E^{-1} \int_0^t A_{11}^T \lambda_1^*(\tau) + \lambda_0^* Q x_1^*(\tau) d\tau.$$

Then we can multiply both sides by A_{21} and using (3.5) we have

$$-A_{21} E^{-1} A_{21}^T \lambda_2^*(t) = A_{21} E^{-1} \int_0^t A_{11}^T \lambda_1^*(\tau) + Q x_1^*(\tau) d\tau. \tag{3.7}$$

By Lemmas 1 and 2, we have that $A_{21}E^{-1}A_{21}^T$ is invertible and we can multiply (3.7) by $-\left(A_{21}E^{-1}A_{21}\right)^{-1}$ and obtain

$$\lambda_2^*(t) = -\left(A_{21}E^{-1}A_{21}^T\right)^{-1} A_{21}E^{-1}\int_0^t A_{11}^T\lambda_1^*(\tau) + \lambda_0^*Qx_1^*(\tau)d\tau.$$

Each component of $\lambda_2^*(t)$ is a linear combination of absolutely continuous functions and consequently $\lambda_2(\cdot)$ is absolutely continuous. Therefore $A_{21}^T\lambda_2^*(t)$ is absolutely continuous and by (3.6) $\lambda_1^*(t)$ is absolutely continuous. Thus, we have the differential algebraic equation

$$-E\dot{\lambda}_1^*(t) = A_{11}^T\lambda_1^*(t) + \lambda_0^*Qx_1^*(t) + A_{21}^T\dot{\lambda}_2^*(t) \tag{3.8}$$

$$A_{21}\lambda_1^*(t) = 0 \tag{3.9}$$

with final condition $\lambda_1^*(T) = 0$. This is the equivalent of the co-state equation for the purely differential problems.

The next question that naturally comes about is the regularity of the problem or, in other words, can λ_0 be equal to 0? Suppose the control problem is not regular or in other words $\lambda_0^* = 0$. Then the system (3.8) and (3.9) becomes

$$-E\dot{\lambda}_1^*(t) = A_{11}^T\lambda_1^*(t) + A_{21}^T\dot{\lambda}_2^*(t) \tag{3.10}$$

$$A_{21}\lambda_1^*(t) = 0.$$

This system satisfies the structure of Lemma 7 and therefore it is well posed. Given the final conditions $\lambda_1(T) = 0$, the solution to (3.10) will be $\lambda_1^*(t) \equiv 0$ and $\dot{\lambda}_2^*(t) \equiv 0$. Since $\lambda_2^*(\cdot)$ is absolutely continuous, $\lambda_2^*(\cdot) \equiv 0$, therefore both $\lambda_0^* = 0$ and $\lambda^* = 0$, which contradicts the Lagrange Multiplier Theorem. Therefore, $\lambda_0^* \neq 0$ and we have a regular problem.

We can take the co-state equation (3.8) and normalize it by λ_0^*, together with the final conditions, we can determine $\lambda_1^*(\cdot)$ and $\lambda_2^*(\cdot)$. We obtained this from the fact that the optimal trajectory is a stationary point of the Lagrangian. The Lagrangian is also stationary in $u^*(\cdot)$. Therefore, we can take the variation of $L(\cdot)$ in $u(\cdot)$ in the direction of $s(\cdot)$ and obtain

$$\int_0^T u(t)^T Rs(t)dt - \int_0^T d\lambda_1^*\left(\int_0^t Bs(\tau)d\tau\right) = 0$$

for all $s(\cdot) \in U$. Integrating the second term by parts, and since $\lambda_1^*(T) = 0$

$$\int_0^T u(t)^T Rs(t)dt + \int_0^T \lambda_1^{T*}(t)Bs(t)dt = 0.$$

Since this is true for all integrable $s(\cdot)$, by (FLCV)

$$u(t)^T R + \lambda_1^{T*}(t)B = 0.$$

Since R is invertible

$$u(t) = -R^{-1}B^T\lambda_1^*(t) \tag{3.11}$$

for almost all $t \in [0, T]$.

For convenience in notation, we can let $\lambda_1(t) = \lambda_1^*(t)$ and $\lambda_2(t) = \dot{\lambda}_2^*(t)$, then we have the co-state system

$$-E\dot{\lambda}_1(t) = A_{11}^T\lambda_1(t) + Qx_1(t) + A_{21}^T\lambda_2(t), \tag{3.12}$$

$$A_{21}\lambda_1(t) = 0, \tag{3.13}$$

$$u(t) = -R^{-1}B^T\lambda_1(t). \tag{3.14}$$

Note that since $x_1(t)$ is smooth, the conditions of Lemma 7 an Theorem 1 are satisfied, therefore we have a well posed equation for $\lambda_1(\cdot)$. Together with (3.1) we have a system of boundary value problems. Since we have an explicit expression for $u(\cdot)$, we can substitute it in (3.1) and obtain

$$E\dot{x_1}(t) = A_{11}x_1(t) + A_{21}^Tx_2(t) - BR^{-1}B^T\lambda_1(t) \tag{3.15}$$

$$A_{21}x_1(t) = 0. \tag{3.16}$$

Boundary conditions are $x_1(0) = x_1^0$ and $\lambda_1(T) = 0$. The system can be written in matrix form

$$\begin{pmatrix} E & 0 & 0 & 0 \\ 0 & -E & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \dot{x}_1(t) \\ \dot{\lambda}_1(t) \\ \dot{x}_2(t) \\ \dot{\lambda}_2(t) \end{pmatrix} = \begin{pmatrix} A_{11} & -BR^{-1}B^T & A_{21}^T & 0 \\ Q & A_{11}^T & 0 & A_{21}^T \\ A_{21} & 0 & 0 & 0 \\ 0 & A_{21} & 0 & 0 \end{pmatrix} \begin{pmatrix} x_1(t) \\ \lambda_1(t) \\ x_2(t) \\ \lambda_2(t) \end{pmatrix}. \tag{3.17}$$

If we substitute t in the second and fourth equation by $T - t$, the structure of the above system is consistent with the structure of Lemma 7. Therefore, the above system is well posed.

3.2 Riccati Equation

In this section we wish to derive an explicit relation between $x_1(\cdot)$ and $\lambda_1(\cdot)$ so that the optimal control $u(\cdot)$ can be expressed explicitly as feedback control.

Conditions (3.13) and (3.16) simply force $\lambda_1(\cdot)$ and $x_1(\cdot)$ to be in the kernel of A_{21}. Thus we can introduce a change of variable. Let the columns of V form an orthonormal basis for $ker(A_{21})$, so that $V^TV = I$. Using V and given $\lambda_1(\cdot)$ and $x_1(\cdot)$, there are unique $z(t)$ and $\xi(t)$ so that

$$\lambda_1(t) = Vz(t) \qquad \text{and} \qquad x_1(t) = V\xi(t).$$

We substitute these expressions in (3.12) and (3.15). Then

$$-EV\dot{z}(t) = A_{11}^T V z(t) + QVz(t) + A_{21}^T \lambda_2(t)$$

$$EV\dot{\xi}(t) = A_{11} V \xi(t) + A_{21}^T x_2(t) - BR^{-1}B^T V z(t).$$

If this system holds, then we can multiply both equations by V^T and obtain a new system.

According to Lemma 3, when we multiply the boundary system by V^T we have the new system

$$-V^T EV\dot{z}(t) = V^T A_{11}^T V z(t) + V^T QVz(t)$$

$$V^T EV\dot{\xi}(t) = V^T A_{11} V \xi(t) - V^T BR^{-1}B^T V z(t).$$

For the system above there is a continuous differentiable operator $\hat{\Pi}(\cdot)$ so that $z(t) = \hat{\Pi}(t)V^T EV\xi(t)$ and $\Pi(t)$ is symmetric positive semi-definite for all t [10]. Then we can rewrite this as a relationship between $\lambda_1(\cdot)$ and $x_1(\cdot)$.

$$z(t) = \hat{\Pi}(t)V^T EV\xi(t) \qquad \Rightarrow \qquad \lambda_1(t) = V\hat{\Pi}(t)V^T E x_1(t).$$

To simplify notation, we multiply by E

$$E\lambda_1(t) = EV\hat{\Pi}(t)V^T E x_1(t).$$

Therefore, we can let $\Pi(t) = EV\hat{\Pi}(t)V^T E$ and thus we have the continuous differentiable $\Pi(\cdot)$ so that $E\lambda_1(t) = \Pi(t)x_1(t)$. Since $\hat{\Pi}(t)$ is symmetric positive semi-definite, $\Pi(t)$ is symmetric positive semi-definite.

The next natural question is the uniqueness of $\Pi(\cdot)$. Both $x_1(t)$ and $\lambda_1(t)$ are vectors in $\mathbf{R}^n$, however, conditions (3.13) and (3.16) restrict $x_1(t)$ and $\lambda_1(t)$ to a subspace of $\mathbf{R}^n$, namely the kernel of A_{21}. For the purpose of the control problem, we are only interested in the action of $\Pi(t)$ on that subspace, the action of $\Pi(t)$ on any vector orthogonal to $ker(A_{21})$ is arbitrary. Given any $\Pi(\cdot)$ so that

$$E\lambda_1(t) = \Pi(t)x_1(t),$$

we can give the optimal feedback control $u(t) = -K(t)x_1(t)$, where $K(t) = R^{-1}B^T E^{-1}\Pi(t)$. Thus, $\Pi(\cdot)$ is not unique and it is easy to see that the general $\Pi(\cdot)$ does not have to be differentiable.

For practical purposes, however, we look for the $\Pi(t)$ with minimum norm. The minimum norm $\Pi(\cdot)$ maps any vector in $ker(A_{21})^{\perp}$ to 0. In a physical system, we usually have an approximation to the actual state. Ideally we wish to distinguish between the error in measurement and the actual state. From the DAE system we know that the algebraic constraint $A_{21}x_1(t) = 0$ is satisfied at every time t, therefore, if we read that the actual state is $\hat{x}_1(t)$, then any part of $\hat{x}_1(t)$ that is orthogonal to $ker(A_{21})$, comes from an error in measurement. We wish to find the $\Pi(\cdot)$ that will simply ignore any such error. We can show that from any solution $\Pi(\cdot)$, we can construct the minimum norm solution in the following way.

Lemma 8 ***Minimum Norm Lemma***

If the columns of V form an orthonormal basis for $ker(A_{21})$ and if $\Pi(\cdot)$ is such that $E\lambda_1(t) = \Pi(t)x_1(t)$, where $x_1(\cdot)$ and $\lambda_1(\cdot)$ satisfy the state and co-state equations (3.17), the minimum norm $\Pi^*(\cdot)$ is given by $\Pi^*(t) = \Pi(t)VV^T$.

***Proof**: By the construction of V, if $x \perp ker(A_{21})$, then $\Pi^*(t)x = 0$ for all t. Therefore, $\Pi^*(\cdot)$ satisfies the condition for minimum norm. It remains to show that $E\lambda_1(t) = \Pi^*(t)x_1(t)$. Suppose $x \in ker(A_{21})$, then by Lemma 6 $V^TVx = x$, therefore*

$$\Pi^*(t)x_1(t) = \Pi(t)VV^Tx_1(t) = \Pi(t)x_1(t) = E\lambda_1(t).$$

According to the above lemma, if we have a $\Pi(\cdot)$, we can always theoretically compute the minimum norm solution. In addition, a corollary of Lemma 8 is that the minimum norm $\Pi(\cdot)$ is differentiable.

We wish to transform conditions (3.12), (3.13), (3.15) and (3.16) into a necessary condition for $\Pi(\cdot)$. Given that $E\lambda_1(t) = \Pi(t)x_1(t)$, we differentiate both sides to obtain

$$E\dot{\lambda}_1(t) = \dot{\Pi}(t)x_1(t) + \Pi(t)\dot{x}_1(t).$$

Substituting the state and co-state equations we have

$$-\dot{\Pi}(t)x_1(t) = A_{11}^TE^{-1}\Pi(t)x_1(t) + \Pi(t)E^{-1}A_{11}x_1(t)$$

$$-\Pi(t)E^{-1}BR^{-1}B^TE^{-1}\Pi(t)x_1(t) + \Pi(t)E^{-1}A_{21}^Tx_2(t) + A_{21}^T\lambda_2(t) + Qx_1(t),$$

$$A_{21}x_1(t) = 0 \qquad \text{and} \qquad A_{21}E^{-1}\Pi(t)x_1(t) = 0.$$

Using the condition that $\lambda_1(T) = 0$, we impose the final condition $\Pi(T) = 0$. The above equation is true for any $x_1(\cdot)$ with corresponding $\lambda_2(\cdot)$ and $x_2(\cdot)$. Because of the algebraic restriction to $x_1(t)$ we cannot take the above expression for any arbitrary vector in $\mathbf{R}^n$, but only for those in $ker(A_{21})$. Given the basis V, we can express $x_1(t)$ as $x_1(t) = V\xi(t)$. Then since $\xi(\cdot)$ is arbitrary, we can write

$$-\dot{\Pi}(t)V\xi(t) = A_{11}^TE^{-1}\Pi(t)V\xi(t) + \Pi(t)E^{-1}A_{11}V\xi(t)$$

$$-\Pi(t)E^{-1}BR^{-1}B^TE^{-1}\Pi(t)V\xi(t) + \Pi(t)E^{-1}A_{21}^Tx_2(t) + A_{21}^T\lambda_2(t) + QV\xi(t),$$

$$A_{21}E^{-1}\Pi(t)V\xi(t) = 0.$$

The other two terms of the equation $x_2(\cdot)$ and $\lambda_2(\cdot)$ depend implicitly on $\xi(\cdot)$. We can solve for $x_2(\cdot)$ from (3.13) and (3.15) and differentiate the algebraic constraint

$$A_{21}x_1(t) = 0 \qquad \Rightarrow \qquad A_{21}\dot{x}_1(t) = 0.$$

Then given $\xi(t)$, we can write the two equations as

$$\begin{pmatrix} E & -A_{21}^T \\ A_{21} & 0 \end{pmatrix}\begin{pmatrix} D \\ x_2(t) \end{pmatrix} = \begin{pmatrix} A_{11}V\xi(t) - BR^{-1}B^TE^{-1}\Pi(t)V\xi(t) \\ 0 \end{pmatrix}.$$

In this case D corresponds to $\dot{x}_1(\cdot)$, but it practically is a redundant variable since it does not explicitly appear anywhere else in the equations. We can solve for D and have the equation for $x_2(t)$

$$0 = A_{21}E^{-1}A_{11}V\xi(t) + A_{21}E^{-1}A_{21}^Tx_2(t) - A_{21}BR^{-1}B^TE^{-1}\Pi(t)V\xi(t).$$

According to Lemma 2, $A_{21}E^{-1}A_{21}^T$ is invertible and therefore we can explicitly solve for $x_2(t)$,

$$\begin{aligned} x_2(t) = & - (A_{21}E^{-1}A_{21}^T)^{-1}A_{21}E^{-1}A_{11}V\xi(t) \\ & + (A_{21}E^{-1}A_{21}^T)^{-1}A_{21}BR^{-1}B^TE^{-1}\Pi(t)V\xi(t). \end{aligned}$$

Combining the equations above and since $\xi(\cdot)$ is arbitrary, we have the system

$$-\dot{\Pi}(t)V = A_{11}^TE^{-1}\Pi(t)V + \Pi(t)E^{-1}A_{11}V$$

$$-\Pi(t)E^{-1}A_{21}^T(A_{21}E^{-1}A_{21}^T)^{-1}A_{21}E^{-1}A_{11}V$$

$$+\Pi(t)E^{-1}A_{21}^T(A_{21}E^{-1}A_{21}^T)^{-1}A_{21}BR^{-1}B^TE^{-1}\Pi(t)V$$

$$-\Pi(t)E^{-1}BR^{-1}B^TE^{-1}\Pi(t)V + A_{21}^T\Lambda(t) + QV,$$

$$0 = A_{21}E^{-1}\Pi(t)V.$$

We can set $\lambda_2(t) = \Lambda(t)\xi(t)$. We can do the same transformation we used for $x_2(\cdot)$ and obtain an explicit form for $\Lambda(t)$ in terms of $\xi(t)$ and $\Pi(t)$, however, that will be impractical. The Riccati Equation at this stage is too complicated to solve in an efficient numerical way. However, we can use the above equation as an analytical tool. For this analytical purpose, we do not need to express $\Lambda(t)$ explicitly.

We do not have enough equations to solve for the system above. The reason for that is that any differentiable $\Pi(\cdot)$ that gives $E\lambda_1(t) = \Pi(t)x_1(t)$, will be a solution to the above system. However, we look for a particular $\Pi(t)$ that will give us the minimum norm solution. Therefore, if we let $\bar{V}$ be a basis for $ker(A_{21})^{\perp}$, we want

$$\Pi(t)\bar{V} = 0.$$

This completes the system with a sufficient number of equations.

We have a non-linear DAE Riccati system of equations and we wish to explore well posedness of the system. From the argument about the change of variable in the co-state system, we know that there exists at least one solution to the Riccati DAE without the minimum norm term. Furthermore, every solution to the above system, gives a $\Pi(\cdot)$, such that $E\lambda_1(t) = \Pi(t)x_1(t)$ is a solution to the state co-state system. We can verify this by substitution. With the addition of the last equation, the above system gives us the unique minimum norm solution. Therefore, the Riccati DAE system is well-posed for any finite time interval $[0, T]$.

In a special case, the Riccati DAE can be simplified. If we assume that $\Pi(\cdot)$ is symmetric, then we observe the following.

Lemma 9 ***Range Relation***

$$If \quad A_{21}E^{-1}\Pi x_1 = 0 \quad \forall x_1 \in ker(A_{21}) \quad and \quad \Pi \quad is \quad symmetric,$$

$$then \quad range(\Pi E^{-1}A_{21}^T) \subseteq range(A_{21}).$$

Proof: *Let $b \in range(\Pi E^{-1}A_{21}^T)$, then $b = \Pi E^{-1}A_{21}^T\xi$ for some ξ. If $x \in ker(A_{21})$, then*

$$\langle b, x\rangle = \left\langle \Pi E^{-1}A_{21}^T\xi, x\right\rangle = \left\langle \xi, A_{21}E^{-1}\Pi x\right\rangle = 0.$$

Therefore, $range(\Pi E^{-1}A_{21}^T)\perp ker(A_{21})$, and by Lemma 4 we can conclude that $range(\Pi E^{-1}A_{21}^T) \subseteq range(A_{21}^T)$.

Therefore, given $S(t)$ and symmetric $\Pi(t)$, there exists $\hat{S}(t)$ so that

$$\Pi(t)E^{-1}A_{21}^T S(t) = A_{21}^T\hat{S}(t).$$

Then we can let $\hat{\Lambda}(t) = \hat{S}(t) + \Lambda(t)$ and simplify the Riccati DAE to

$$\begin{aligned}
-\dot{\Pi}(t)V &= A_{11}^T E^{-1}\Pi(t)V + \Pi(t)E^{-1}A_{11}V \\
&\quad -\Pi(t)E^{-1}BR^{-1}B^T E^{-1}\Pi(t)V + A_{21}^T\hat{\Lambda}(t) + QV, \\
0 &= A_{21}E^{-1}\Pi(t)V.
\end{aligned}$$

The above system is smaller and simpler, however, it gives the necessary condition for Π only if Π is symmetric. We know there is always a symmetric solution $\Pi(\cdot)$, however, the minimum norm solution does not have to be symmetric. Unless the minimum norm $\Pi(\cdot)$ is symmetric, we cannot complete the above system with the term $\Pi(t)\bar{V} = 0$ and we cannot create a well-posed system.

We are interested in particular in the solution to the optimal regulator problem. The regulator problem can be expressed as the limit of the finite time minimization problem as $T \to \infty$. The optimal Π for that problem will be constant in time and satisfy [5, 11]

$$\Pi = \lim_{t \to -\infty} \Pi(t),$$

where $\Pi(\cdot)$ is the solution to the Riccati equation with final condition $\Pi(0) = 0$. If Π exists it will satisfy

$$A_{21}^T \Lambda = A_{11}^T E^{-1} \Pi V + \Pi E^{-1} A_{11} V \tag{3.18}$$

$$-\Pi E^{-1} A_{21}^T (A_{21} E^{-1} A_{21}^T)^{-1} A_{21} E^{-1} A_{11} V$$

$$+\Pi E^{-1} A_{21}^T (A_{21} E^{-1} A_{21}^T)^{-1} A_{21} B R^{-1} B^T E^{-1} \Pi V,$$

$$-\Pi E^{-1} B R^{-1} B^T E^{-1} \Pi V + QV$$

$$A_{21} E^{-1} \Pi V = 0. \tag{3.19}$$

If Π is symmetric it is enough to consider

$$-A_{21}^T \Lambda = A_{11}^T E^{-1} \Pi V + \Pi E^{-1} A_{11} V - \tag{3.20}$$

$$-\Pi E^{-1} B R^{-1} B^T E^{-1} \Pi V + QV$$

$$A_{21} E^{-1} \Pi V = 0. \tag{3.21}$$

Furthermore, we wish for Π to be of minimum norm, so that $\Pi\bar{V} = 0$.

The algebraic Riccati equation is non-linear and therefore it can possibly have multiple solutions. Only one of those solutions will be the limit of the Riccati DAE as $t \to -\infty$ (if that limit exists). If we have a regular Riccati equation, we know that the solution has to be symmetric positive semi-definite, however, in our case we cannot make any such assumptions. Therefore, even if we can find all the solutions to the above system, we cannot create a condition that will allow us to select the desired solution.

Neither the Riccati DAE nor the Algebraic Riccati Equations are practical to solve for large systems. Therefore, we wish to try and approximate Π, by a solution to a regular Riccati equation.

3.3 Perturbation of the System

The first alternative way of finding Π is to perturb the original DAE system. Given some small $\epsilon > 0$ and easily invertible matrix M, we perturb (3.1) to:

$$\begin{pmatrix} E & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{pmatrix} = \begin{pmatrix} A_{11} & A_{21}^T \\ A_{21} & \epsilon M \end{pmatrix} \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} + \begin{pmatrix} B \\ 0 \end{pmatrix} u_\epsilon(t) \qquad (3.22)$$

Then the system can be reduced to the purely differential form:

$$E\dot{x}_1(t) = (A_{11} - \frac{1}{\epsilon} A_{21}^T M^{-1} A_{21}) x_1(t) + B u_\epsilon(t)$$

or equivalently:

$$\dot{x}_1(t) = (E^{-1}A_{11} - \frac{1}{\epsilon} E^{-1} A_{21}^T M^{-1} A_{21}) x_1(t) + E^{-1} B u_\epsilon(t)$$

Note that our choice for the structure of $J9\cdot)$ gives us the familiar

$$J_\epsilon(u(\cdot)) = \frac{1}{2} \int_0^\infty \langle x_1(t), Q x_1(t) \rangle + \langle u_\epsilon(t), R u_\epsilon(t) \rangle \, dt.$$

The properties of control problem of the form (3.22) have been studied in [2, 6]. The control for (3.22) exists and is given by $u_\epsilon(t) = -K_\epsilon x_1(t)$, however, the optimal gain K_ϵ is not unique. Bender and Laub give Riccati Equations for the minimum norm solution, however, for our purpose, the minimum norm gain for (3.22) may not be the minimum norm gain for (3.1). Therefore, we try to find any feedback law for (3.22) and then use Lemma 8.

We can find an optimal Π_ϵ for problem (3.22) by the following. For given value of ϵ we have a matrix Π_ϵ that satisfies:

$$\begin{aligned} F^T(\epsilon)\Pi_\epsilon &+ \Pi_\epsilon F(\epsilon) - \Pi_\epsilon E^{-1} B R^{-1} B^T E^{-1} \Pi_\epsilon + Q = 0, \\ \text{where } F(\epsilon) &= E^{-1} A_{11} - \frac{1}{\epsilon} E^{-1} A_{21}^T M^{-1} A_{21}. \end{aligned}$$

We will show that if $\Pi = \lim_{\epsilon \to 0} \Pi_\epsilon$ exists, then Π satisfies the Riccati equation (3.18), (3.19).

The existence of the limit depends upon the choice of M. In the case of the Stokes problem, the divergence condition in the PDE itself could be perturbed to

$$\nabla \cdot \vec{u} + \epsilon p = 0.$$

Thus, Galerkin finite element approximation lead to a symmetric positive definite mass matrix M and in numerical experiments $\Pi_\epsilon \to \Pi$. In other experiments, $M = I$, was chosen and still convergence was still achieved. However, not every choice of M gives convergence. If we perturb the Stokes problem to $\nabla \cdot \vec{u} - \epsilon p = 0$, then Π_ϵ diverges. The convergent perturbation to the Stokes problem corresponds to adding artificial diffusion, while the divergent one adds negative diffusion, which has no physical meaning. Therefore, the choice of M is not arbitrary and has to be made based on insight from the specific problem.

Convergence analysis for the perturbed system is given in Section 3.5.

3.4 Change of Variable

Another way to handle the problem is to perform a change of variables. The constraint $A_{21}x_1(t) = 0$ simply means that at any time $x_1(t) \in ker(A_{21})$. Thus, if the columns of matrix V form an orthonormal basis for the $ker(A_{21})$, at each time, we can express $x_1(t)$ uniquely by $x_1(t) = V\xi(t)$ for some $\xi(t)$. We can substitute this decomposition in the original system for $x_1(t)$,

$$EV\dot{\xi}(t) = A_{11}V\xi(t) + A_{21}^T x_2(t) + Bu(t).$$

We can multiply both sides of the equation by V^T and obtain

$$V^T EV\dot{\xi}(t) = V^T A_{11}V\xi(t) + V^T A_{21}^T x_2(t) + V^T Bu(t).$$

By Lemma 3, $V^T A_{21}^T = 0$, therefore, we have the system,

$$V^T EV\dot{\xi}(t) = V^T A_{11}V\xi(t) + V^T Bu(t), \tag{3.23}$$

for the new variable $\xi(\cdot)$. Initial conditions can be obtained by observing that $x_1(0) \in ker(A_{21})$. Thus our consistency assumption implies $x_1(t) = V\xi(t)$, then $\xi(0) = V^T x_1(0)$.

On this equation we can impose a corresponding cost functional:

$$\begin{aligned} J_\xi(u(\cdot)) &= \frac{1}{2}\int_0^\infty \langle V\xi(t), QV\xi(t)\rangle + \langle u(t), Ru(t)\rangle \, dt \\ &\text{or} \\ J_\xi(u(\cdot)) &= \frac{1}{2}\int_0^\infty \left\langle \xi(t), V^T QV\xi(t)\right\rangle + \langle u(t), Ru(t)\rangle \, dt. \end{aligned}$$

Since E is positive definite and columns of V form an irreducible basis, from Lemma 2 $V^T EV$ is invertible. Therefore the system (3.23) is purely differential and since the weight in the cost functional, $V^T QV$, is also symmetric positive semi-definite, the optimal control $u_\xi(t)$ can be found of the form [11, 5, 10]

$$u_\xi(t) = -R^{-1}B^TV(V^TEV)^{-1}\Gamma\xi(t),$$

where Γ is symmetric positive semi-definite matrix that satisfies the Riccati equation

$$\begin{gathered}V^TA_{11}^TV(V^TEV)^{-1}\Gamma + \Gamma(V^TEV)^{-1}V^TA_{11}V - \\ -\Gamma(V^TEV)^{-1}V^TBR^{-1}B^TV(V^TEV)^{-1}\Gamma + V^TQV = 0.\end{gathered}$$

The control computed in this way will be in terms of the new variable $\xi(t)$ and in order to convert it to control in terms of the original variable $x_1(t)$ we use the fact that V is orthogonal and $\xi(t) = V^Tx_1(t)$. Therefore

$$u(t) = R^{-1}B^TE^{-1}EV(V^TEV)^{-1}\Gamma V^Tx_1(t).$$

So the matrix Π_ϵ for the original system is given by

$$\Pi_\epsilon = EV(V^TEV)^{-1}\Gamma V^T.$$

Proof that the above Π_ϵ will give an optimal control for the original DAE system (3.1) (i.e. $u(t) = u_\epsilon(t)$), is given in Section 3.6.

3.5 Convergence of the Perturbed System

First we consider the finite time case. Suppose M was chosen so that M is symmetric positive definite, $\Pi_\epsilon(t) \to \Pi(t)$ and $\Pi(\cdot)$ is continuous for all $t \in [0, T]$. Since all $\Pi_\epsilon(\cdot)$ are continuous and since they converge pointwise on a compact domain, they converge uniformly. If we let $T \to \infty$ and if the perturbed systems are stabilizable, then $\Pi_\epsilon(t)$ will asymptotically converge to some Π_ϵ and therefore we can conclude that $\Pi_\epsilon(\cdot) \to \Pi(\cdot)$ uniformly on $(-\infty, 0]$. We assume that $\Pi_\epsilon(\cdot) \to \Pi(\cdot)$ uniformly and we want to show that for any fixed t, $\Pi(t)$ will satisfy the Riccati DAE given in Section 3.2.

Fix t and then consider the limit $\Pi(t) = \lim_{\epsilon\to 0}\Pi_\epsilon(t)$.

Lemma 10 ***Symmetric Positive Semi-Definite***

$$\Pi(t) = \lim_{\epsilon\to 0}\Pi_\epsilon(t) \textit{ is symmetric positive semi-definite.}$$

Proof: *Since each of the* $\Pi_\epsilon(t)$ *is symmetric, symmetry is trivial.*

$$\Pi^T(t) = \lim_{\epsilon\to 0}\Pi_\epsilon^T(t) = \lim_{\epsilon\to 0}\Pi_\epsilon(t) = \Pi(t).$$

Definiteness involves a little more work.
Let $\eta > 0$ and $x \in \boldsymbol{R}^n$ be arbitrary. Define

$$\Delta\Pi_\epsilon = \Pi(t) - \Pi_\epsilon(t),$$

then $\Delta\Pi_\epsilon(t) \to 0$ as $\epsilon \to 0$. There exists $\delta > 0$ so that $\|\Delta\Pi_\epsilon\| < \frac{\eta}{\|x\|^2}$, whenever $|\epsilon| < \delta$. Thus if $0 < \epsilon^* < \delta$

$$\begin{aligned}\langle \Pi(t)x, x\rangle &= \langle \Pi_\epsilon^*(t)x, x\rangle + \langle \Delta\Pi_\epsilon^*(t)x, x\rangle \\ &\geq 0 - \frac{\eta}{\|x\|^2}\|x\|^2 = -\eta.\end{aligned}$$

Since η was arbitrary $\langle \Pi(t)x, x\rangle \geq 0$ and since x was arbitrary $\Pi(t)$ is positive semi-definite. Thus $\Pi(t)$ is symmetric positive semi-definite for any value of $t \in [0, T]$.

We know that $\Pi_\epsilon(T) = 0$ for all ϵ and we know that for all $t \in [0, T]$, $\Pi_\epsilon(t)$ satisfies the Riccati Differential Equation

$$\begin{aligned}\dot{\Pi}_\epsilon(t) + F_\epsilon^T\Pi_\epsilon(t) &+ \Pi_\epsilon(t)F_\epsilon - \Pi_\epsilon(t)E^{-1}BR^{-1}B^TE^{-1}\Pi_\epsilon(t) + Q = 0 \\ \text{where } F_\epsilon &= E^{-1}A_{11} - \frac{1}{\epsilon}E^{-1}A_{21}^TM^{-1}A_{21}.\end{aligned}$$

Equivalently:

$$\begin{aligned}&\dot{\Pi}_\epsilon(t) + A_{11}^TE^{-1}\Pi_\epsilon(t) + \Pi_\epsilon(t)E^{-1}A_{11} - \Pi_\epsilon(t)E^{-1}BR^{-1}B^TE^{-1}\Pi_\epsilon(t) + Q \\ &\qquad = \frac{1}{\epsilon}\left(A_{21}^TM^{-1}A_{21}E^{-1}\Pi_\epsilon(t) + \Pi_\epsilon(t)E^{-1}A_{21}^TM^{-1}A_{21}\right).\end{aligned}$$

If we take the limit on both sides, we have

$$\begin{aligned}&\dot{\Pi}(t) + A_{11}^TE^{-1}\Pi(t) + \Pi(t)E^{-1}A_{11} - \Pi(t)E^{-1}BR^{-1}B^TE^{-1}\Pi(t) + Q \\ &\qquad = \lim_{\epsilon\to 0}\frac{1}{\epsilon}\left(A_{21}^TM^{-1}A_{21}E^{-1}\Pi_\epsilon(t) + \Pi_\epsilon(t)E^{-1}A_{21}^TM^{-1}A_{21}\right).\end{aligned}$$

The minimum norm solution to the Riccati DAE does not have to be symmetric. Since the limit $\Pi(\cdot)$ is always symmetric it may not be the minimum norm solution, however, we wish to show that the limit is a solution to the Riccati DAE. Because of symmetry of $\Pi(\cdot)$ it is enough to consider the following two equations

$$-A_{21}^T\lambda_2 = (\dot{\Pi}(t) + A_{11}^TE^{-1}\Pi(t) + \Pi(t)E^{-1}A_{11} \tag{3.24}$$

$$-\Pi(t)E^{-1}BR^{-1}B^TE^{-1}\Pi(t) + Q)x_1,$$

$$A_{21}E^{-1}\Pi(t)x_1 = 0, \tag{3.25}$$

for all $x_1 \in ker(A_{21})$. In other words, we want to show that for any $x_1 \in ker(A_{21})$, there exists λ_2 so that (3.24) and (3.25) hold.

A necessary condition for

$$\lim_{\epsilon \to 0} \frac{1}{\epsilon} \left(A_{21}^T M^{-1} A_{21} E^{-1} \Pi_\epsilon(t) + \Pi_\epsilon(t) E^{-1} A_{21}^T M^{-1} A_{21} \right) = S(t) \qquad (3.26)$$

to exist is that

$$\lim_{\epsilon \to 0} \left(A_{21}^T M^{-1} A_{21} E^{-1} \Pi_\epsilon + \Pi_\epsilon E^{-1} A_{21}^T M^{-1} A_{21} \right) = 0,$$

thus $A_{21}^T M^{-1} A_{21} E^{-1} \Pi(t) + \Pi(t) E^{-1} A_{21}^T M^{-1} A_{21} = 0$.

Lemma 11 ***Kernel Invariance***

$$\textit{If } A_{21}^T M^{-1} A_{21} E^{-1} \Pi(t) + \Pi(t) E^{-1} A_{21}^T M^{-1} A_{21} = 0 \textit{ then}$$

$$E^{-1} \Pi(t) x_1 \in ker(A_{21}) \quad \forall x_1 \in ker(A_{21}).$$

Proof: *Let* $x_1 \in ker(A_{21})$, *then*

$$\left(A_{21}^T M^{-1} A_{21} E^{-1} \Pi(t) + \Pi(t) E^{-1} A_{21}^T M^{-1} A_{21} \right) x_1 = 0.$$

But $\Pi(t) E^{-1} A_{21}^T M^{-1} A_{21} x_1 = 0$, *therefore,*

$$A_{21}^T M^{-1} A_{21} E^{-1} \Pi(t) x_1 = 0.$$

Let $c = E^{-1} \Pi(t) x_1$. *Then consider*

$$0 = \left\langle A_{21}^T M^{-1} A_{21} c, c \right\rangle = \left\langle M^{-1} A_{21} c, A_{21} c \right\rangle.$$

Since M *is positive definite* $A_{21} c = 0$, *therefore,*

$$c = E^{-1} \Pi(t) x_1 \in ker(A_{21}) \quad \forall x_1 \in ker(A_{21}).$$

According to Lemma 11, the limit $\Pi(t)$ will satisfy condition (3.25). It remains to show that it will satisfy condition (3.24). Condition (3.24) states that for every $x_1 \in ker(A_{21})$ there is a λ_2 so that the equation holds true.

Lemma 12 ***Range of the Limit***

$$\textit{If } \lim_{\epsilon \to 0} \frac{1}{\epsilon} \left(A_{21}^T M^{-1} A_{21} E^{-1} \Pi_\epsilon(t) + \Pi_\epsilon(t) E^{-1} A_{21}^T M^{-1} A_{21} \right) = S(t),$$

then $S(t)x_1 \in range(A_{21}^T)\ \forall x_1 \in ker(A_{21})$.

Proof: *Let $x_1 \in ker(A_{21})$ and let $\epsilon > 0$ be arbitrary. Consider*

$$\frac{1}{\epsilon}\left(A_{21}^T M^{-1} A_{21} E^{-1}\Pi_\epsilon(t) + \Pi_\epsilon(t) E^{-1} A_{21}^T M^{-1} A_{21}\right) x_1.$$

It can be split into two parts. First it is obvious that

$$\frac{1}{\epsilon} A_{21}^T M^{-1} A_{21} E^{-1} \Pi_\epsilon(t) x_1 \in range(A_{21}^T),$$

regardless of the choice of x_1. Second part is

$$\frac{1}{\epsilon}\Pi_\epsilon(t) E^{-1} A_{21}^T M^{-1} A_{21} x_1 = 0 \textit{ since } A_{21}x_1 = 0.$$

Thus, for any arbitrary $\epsilon > 0$

$$\frac{1}{\epsilon}\left(A_{21}^T M^{-1} A_{21} E^{-1}\Pi_\epsilon(t) + \Pi_\epsilon(t) E^{-1} A_{21}^T M^{-1} A_{21}\right) x_1 \in range(A_{21}^T)$$

and because subspaces of $\boldsymbol{R}^n$ are complete, in the limit

$$S(t)x_1 \in range(A_{21}^T) \quad \forall x_1 \in ker(A_{21}).$$

Lemma 12 gives the existence of a λ_2 for all $x_1 \in ker(A_{21})$, therefore, condition (3.24) is satisfied. Combining the two results we know that if $\Pi_\epsilon(\cdot) \to \Pi(\cdot)$ point-wise on $[0, T]$ and if $\Pi(\cdot)$ is continuous, then the limit $\Pi(\cdot)$ will satisfy the Riccati DAE and the convergence will be uniform.

We can consider the regulator problem by letting $T \to \infty$. In this case the Riccati Equations have final conditions $\Pi_\epsilon(0) = 0$ and we will be interested in the limit as $t \to -\infty$. If we assume that all $\Pi_\epsilon(\cdot)$ are bounded, converge point-wise to a continuous $\Pi(\cdot)$ on $(-\infty, 0]$ and $\Pi(\cdot)$ is bounded, then the convergence is uniform and the limit $\Pi(\cdot)$ will satisfy the Riccati DAE. If we let $\Pi_\epsilon = \lim_{t\to-\infty}\Pi_\epsilon(t)$, the optimal control for the DAE will be given by $u(t) = -R^{-1}B^T E^{-1}\Pi x_1(t)$, where

$$\Pi = \lim_{t\to-\infty}\Pi(t) = \lim_{\epsilon\to 0}\Pi_\epsilon.$$

We can find Π_ϵ by directly solving the standard Algebraic Riccati Equation

$$\begin{aligned} F_\epsilon^T \Pi_\epsilon &+ \Pi_\epsilon F_\epsilon - \Pi_\epsilon E^{-1} B R^{-1} B^T E^{-1} \Pi_\epsilon + Q = 0 \\ \text{where } F_\epsilon &= E^{-1} A_{11} - \frac{1}{\epsilon} E^{-1} A_{21}^T M^{-1} A_{21}. \end{aligned}$$

Methods for solving the above equation are discussed in Chapter 4.

Theorem 2 ***Epsilon Convergence***

If $\Pi_\epsilon(\cdot)$ give the optimal feedback control for the perturbed system and if

$$\Pi(t) = \lim_{\epsilon \to 0} \Pi_\epsilon(t)$$

exists, is bounded and continuous on the specified time interval, then $\Pi(\cdot)$ is symmetric positive semi-definite for all t and it satisfies the Riccati DAE equation derived in Section 3.2. However, Π may not be of minimum norm.

3.6 Equivalence of the Change of Variables

Consider the original control problem (3.1) and the Change of Variable control problem (3.23). If we do not look at either Π or Γ, we can give the following result.

Lemma 13 ***Equivalence for the Controls***

$u^*(\cdot)$ is optimal control for the DAE system (3.1) if and only if it is optimal control for Change of Variable system (3.23).

Proof: *Suppose that $u^*(\cdot)$ is an optimal control for DAE (3.1) and not an optimal control for Change of Variable (3.23), then there is $q^*(\cdot)$ so that*

$$J_\xi(q^*(\cdot)) < J_\xi(u^*(\cdot)).$$

Then we can consider the trajectory $x_{1q}(\cdot)$ given by the solution to

$$E\dot{x}_{1q}(t) = A_{11}x_{1q}(t) + A_{21}x_2(t) + Bq^*(t).$$

$$0 = A_{21}x_{1q}(t).$$

Since $q^*(\cdot) \not\equiv u^*(\cdot)$

$$J(u^*(\cdot)) \leq J(q^*(\cdot)).$$

If we let $x_{1u}(\cdot)$ be the solution to (3.1) corresponding to the optimal control $u^*(\cdot)$ and if we let $x_{1q}(t) = V\xi_q(t)$ and $x_{1u}(t) = V\xi_u(t)$, then $\xi_q(\cdot)$ and $\xi_u(\cdot)$ are the trajectories in (3.23) corresponding to $q^*(\cdot)$ and $u^*(\cdot)$. But the functional $J_\xi(\cdot)$ was defined so that $J_\xi(u(\cdot)) = J(u(\cdot))$, therefore

$$J_\xi(q^*(\cdot)) \geq J_\xi(u^*(\cdot)).$$

This is a contradiction and therefore $u^*(\cdot)$ is optimal control for Change of Variable (3.23).

Suppose that $u^*(\cdot)$ is optimal control for Change of Variable (3.23) and not optimal for DAE (3.1), then there is $q^*(\cdot)$ so that

$$J(q^*(\cdot)) < J(u^*(\cdot)).$$

By construction of $J_\xi(\cdot)$, $J_\xi(u(\cdot)) = J(u(\cdot))$, therefore

$$J_\xi(q^*(\cdot)) < J_\xi(u^*(\cdot)).$$

However, $u^*(\cdot)$ is optimal control for (3.23) and thus

$$J_\xi(q^*(\cdot)) \geq J_\xi(u^*(\cdot)).$$

This is a contradiction and therefore $u^*(\cdot)$ is optimal control for DAE (3.1).

If we consider the case with finite time, the new system (3.23) has unique optimal control [11, 5, 10]. Therefore, by Lemma 13 there is unique control to DAE (3.1).

For the regulator problem with $T = \infty$, the change of variable system has a unique control if and only if it is stabilizable [11, 5]. Therefore,DAE (3.1) has a unique control if and only if Change of Variable (3.23) is stabilizable.

The optimal control for (3.23), in the finite time case is given by

$$u_\xi(t) = -R^{-1}B^TV(V^TEV)^{-1}\Gamma(t)\xi(t),$$

where $\Gamma(\cdot)$ is the solution to a Riccati Differential Equation. Given $\Gamma(\cdot)$, we can define $\Pi(t) = EV(V^TEV)^{-1}\Gamma(t)V^T$. Then the optimal control for (3.1) will be given by

$$u(t) = -R^{-1}B^TE^{-1}\Pi(t)x_1(t).$$

By construction of $\Pi(\cdot)$ and by the properties of V, $\Pi(\cdot)$ will be the minimum norm solution.

The optimal control for the Change of Variable system, in the regulator case is given by

$$u(t) = -R^{-1}B^TV(V^TEV)^{-1}\Gamma\xi(t),$$

where Γ is the symmetric positive definite solution to a Riccati Algebraic Equation. The minimum norm Π that will give the control for the DAE system is given by

$$\Pi = EV(V^TEV)^{-1}\Gamma V^T.$$

3.7 Summary of the Optimal Control Results

Given a DAE system of the form (3.1), we know that the optimal control is given by

$$u(t) = -R^{-1}B^T E^{-1}\Pi x_1(t).$$

The optimal Π is not unique, however, there is a minimum norm solution that is the one that we need for practical purposes.

The optimal minimum norm Π can be found from a system of Riccati DAE or Riccati Algebraic equations. Both systems are impractical for two reasons. The first reason is that they require the computation of some redundant terms or involve complicated matrix multiplication so they can be computationally very expensive. The second reason is that there are no good numerical ways developed for finding the solution of an Riccati Algebraic equation of that form.

There are alternative ways to compute the optimal gain. We consider two of these. The first is to perturb the original system by some small ϵ and have an approximate optimal Π. The advantage of this method is that if the matrices involved in the DAE are sparse, the Riccati Equation for the optimal Π can be solved in terms of sparse operations. The first disadvantage is that it only gives an approximation to an optimal Π. The second disadvantage is that even if the approximation converges to a solution of the Riccati DAE, it may not converge to the minimum norm solution.

The second way to compute the optimal Π is to do a change of variables and compute the control for the resulting purely differential system. The advantage of this method is that it gives the exact minimum norm Π. One disadvantage is that a basis for $ker(A_{21})$ has to be computed, which can be computationally very expensive. Another disadvantage is that even if the original system is sparse the new system is always dense and the Riccati equation will have to be solved using dense matrix operations.

Chapter 4

Riccati Solver

4.1 Chandrasekhar Algorithm

4.1.1 Description of the Algorithm

Given a system of differential equations

$$\dot{x}(t) = Ax(t) + Bu(t).$$

A is an $n \times n$ matrix, B is an $n \times m$ matrix, $x(\cdot)$ is a vector function $x : \mathbf{R} \rightarrow \mathbf{R}^n$ and $u(\cdot)$ is a vector function $u : \mathbf{R} \rightarrow \mathbf{R}^m$. We wish to find an optimal control $u^*(\cdot)$ that minimizes the functional

$$J(u(\cdot)) = \int_0^T \langle x(t), Qx(t) \rangle + \langle u(t), Ru(t) \rangle \, dt.$$

We assume Q is a symmetric positive semi-definite matrix and R is a symmetric positive definite matrix. The optimal control has the linear feedback form

$$u^*(t) = -K(t)x(t).$$

The optimal gain $K(\cdot)$ can be determined by $K(t) = R^{-1}B^T\Pi(t)$, where $\Pi(t)$ is the solution to the Differential Riccati Equation (DRE)

$$-\dot{\Pi}(t) = A^T\Pi(t) + \Pi(t)A - \Pi(t)BR^{-1}B^T\Pi(t) + Q, \tag{4.1}$$

with final condition $\Pi(T) = 0$ [5].

The matrix function $\Pi(\cdot)$ is symmetric positive semi-definite at each time t. Thus for practical purposes, if we wish to solve the DRE we need to solve a system of $\frac{n(n-1)}{2}$ differential equations. However, we are only interested in finding the optimal gain $K(\cdot)$, which for many practical applications has a much smaller dimension. Thus, we are actually interested in solving for $m \times n$ functions that will give us the optimal gain $K(\cdot)$. If $m \ll n$, then the DRE solves for *many* more equations than we actually need.

The Chandrasekhar Method for solving DRE takes advantage of the structure of the problem when $m \ll n$ [11, 1, 5]. The method is derived by looking at

$$-K(t) = -R^{-1}B^T\Pi(t).$$

Differentiating both sides we obtain a differential equation for $K(\cdot)$,

$$-\dot{K}(t) = -R^{-1}B^T\dot{\Pi}(t).$$

Final conditions can be obtained from the final conditions for $\Pi(T)$, therefore $K(T) = 0$. Then we need a way to rewrite $\dot{\Pi}(t)$. If we take (4.1) and differentiate once, we have

$$-\ddot{\Pi}(t) = A^T\dot{\Pi}(t) + \dot{\Pi}(t)A - \dot{\Pi}(t)BR^{-1}B^T\Pi(t) - \Pi(t)BR^{-1}B^T\dot{\Pi}(t),$$

which can be factored as

$$-\ddot{\Pi}(t) = \left(A - BR^{-1}B^T\Pi(t)\right)^T\dot{\Pi}(t) + \dot{\Pi}(t)\left(A - BR^{-1}B^T\Pi(t)\right).$$

Since $K(t) = R^{-1}B^T\Pi(t)$, we can substitute

$$-\ddot{\Pi}(t) = (A - BK(t))^T\dot{\Pi}(t) + \dot{\Pi}(t)(A - BK(t)).$$

We can look at the above as a differential equation with final conditions $\dot{\Pi}(T) = Q$. Then we can let $U(\cdot)$ be the solution to the differential equation

$$-\dot{U}(t) = (A - BK(t))^T U(t)$$

with final condition $U(T) = I$. Then

$$-\dot{\Pi}(t) = U(t)QU^T(t),$$

which can be verified by differentiation. The positive definite matrix Q can be factorized into $Q = C^TC$, and we define

$$L(t) = CU^T(t).$$

Therefore $-\dot{\Pi}(t) = L^T(t)L(t)$. Thus we find a differential equation for $L(\cdot)$

$$-\dot{L}(t) = C\dot{U}^T(t) = CU^T(t)(A - BK(t)) = L(t)(A - BK(t)).$$

The final condition can be derived from the initial condition for $U(\cdot)$, therefore $L(T) = C$. In this way, we have developed a system of differential equations

$$-\dot{K}(t) = R^{-1}B^T L^T(t)L(t) \tag{4.2}$$

$$-\dot{L}(t) = L(t)\left(A - BK(t)\right) \tag{4.3}$$

with final conditions

$$K(T) = 0 \qquad L(T) = C. \tag{4.4}$$

The system (4.2), (4.3), (4.4) is called the Differential Chandrasekhar Equation (DCE). The total number of equations to be solved is $(m + p) \times n$, where $p = rank(Q)$. In many practical applications, we are not interested in minimizing all the state variables of $x(\cdot)$, but rather a small number of observations. If $p \ll n$ we have $(m + p) \ll n$. In this case it is significantly more efficient to solve the DCE as opposed to the DRE.

4.1.2 Numerical Solution

In this section we will introduce a good numerical method for solving (4.2), (4.3), (4.4).

Equations (4.2), (4.3) and (4.4) are integrated backwards in time, but for simplicity in notation, we can remove the negative signs before the derivatives and integrate forward in time ($t \rightarrow T - t$). Then given time steps t_i, where $t_0 = 0$ and $t_{i+1} = t_i + \Delta t$, we want to generate sequence of solutions K_i and L_i so that $K_0 = 0$, $L_0 = C$ and $K_i \approx K(t_i)$, $L_i \approx L(t_i)$.

The structure of the DCE introduces many difficulties for most standard ODE integrators. Stability is the main issue for any explicit method. In many practical applications Δt has to be taken very small and the equations take too long to integrate to be practical. If we use an implicit method, we have to solve a system of non-linear equation at each step, which in itself can be very unstable and time consuming.

A very efficient numerical method for solving (4.2), (4.3), (4.4) was proposed by Banks and Ito [1]. The main observation is that $\dot{K}(\cdot)$ only depends on the values of $L(\cdot)$ and that $\dot{L}(\cdot)$ depends linearly on $L(\cdot)$. Given Δt and the approximate solutions K_i and L_i up until t_n, we can use an explicit method to find $\hat{K}_{n+1}$ as a guess for K_{n+1}. Then using $\hat{K}_{n+1}$ we can approximate (4.3) by

$$\dot{L}(t) = L(t)\left(A - B\hat{K}_{n+1}\right).$$

Using the above formula we can take a step with an implicit method to find the next step L_{n+1}. The advantage is that for the implicit step we will only have to solve a linear system. Once we have L_{n+1} we can take an implicit step and find the next iterate K_{n+1}.

The specific explicit and implicit method used at each step can vary. Banks and Ito [1] used second order Adams-Bashford method for the explicit step and second order Adams-Moulton for the implicit steps. The advantage of this combination is very high stability. The disadvantage is that it is only second order. In numerical experiments a combination of fifth order Adams-Bashford Adams-Moulton methods resulted in better accuracy but decreased stability.

4.2 Optimized Newton Method

4.2.1 Chandrasekhar as Initial Guess to Newton

The Chandrasekhar method is very attractive when we wish to solve an optimization problem for finite time, however, it looses some of its advantages, when $T \to \infty$. If the cost functional that we wish to minimize is given by

$$J(u(\cdot)) = \int_0^\infty \langle x(t), Qx(t)\rangle + \langle u(t), Ru(t)\rangle \, dt,$$

then the optimal control is given by $u^*(t) = -Kx(t)$, where

$$K = \lim_{t \to -\infty} K(t).$$

The function $K(\cdot)$ is the solution to (4.2), (4.3), (4.4) with $T = 0$. If we wish to numerically integrate $K(\cdot)$, we need to integrate up until some very large time $\hat{t}$ so that $\|L(\hat{t})\| < tol$. The problem is that both the numerical error (accuracy) and the stability of any method depend on the length of the interval over which we integrate. Therefore, if $\hat{t}$ is very large, low order stable method may give poor accuracy, while high order methods will become unstable.

The solution to this problem, as proposed by Banks and Ito [1], is to integrate $K(\cdot)$ until some moderate $\hat{t}$ and then use $K(\hat{t})$ as initial guess for Newton iteration. The Newton iteration for the Algebraic Riccati Equation is guaranteed to converge quadratically to the solution, if $(A - BK_0)$ is stable [1, 5]. The algorithm uses K_0 as initial guess, which is obtained from partially integrating the DCE. On each step of the Newton iteration, the solution to a Lyapunov system is required.

The main disadvantage of this method is the cost of finding the solution of Lyapunov Equations. Factorization of an $n \times n$ dense matrix is required at every step and the solution is a large matrix Π. Banks and Ito propose an alternative iteration, which is derived from the standard one, but generates a sequence of iterates for the gain K_i and only requires the solution to several linear systems. The only computationally expensive part of that algorithm is the solution to a linear system of the form $X\left(I - rA + rBK_i\right) = RHS$, where r is a step size prescribed to speed convergence. There are many efficient methods to solve linear systems. Banks and Ito used standard LU factorization, however, if the size of the system is too large, iterative methods could give better performance.

4.2.2 Sparse Systems

Large control systems usually come from the discretization of PDE system using some form of finite difference or finite element method. In those cases the matrix A would be either sparse or $A = E^{-1}\hat{A}$, where both E and $\hat{A}$ are sparse. Since the algorithm proposed by Banks and Ito uses only linear solvers we can take advantage of the sparsity.

In both the integration of the Chandrasekhar system and the Newton iteration that follows, we have to solve a system of equations of the form

$$X\left(I - dA + dB\hat{K}\right) = RHS.$$

$\hat{K}$ is some approximation to the optimal gain, d depends on Δt or the size of the step in the Lyapunov solver and RHS is some matrix on the right hand side of the equation. The number of equations that need to be solved depends on the size of $L(\cdot)$ in DCE or the size of K in the Newton iteration. For simplicity of the argument, we can assume that we have only one equation. Therefore, if A is sparse and if B is a thin matrix (i.e. $m \ll n$), then the action of $\left(I - dA + dB\hat{K}\right)$ onto X can be found very efficiently using $O(nm)$ number of operations. If $A = E^{-1}\hat{A}$ then we can rewrite the equation as:

$$Z\left(E - d\hat{A} + dEB\hat{K}\right) = RHS.$$

The solution X will be given by $X = ZE$.

A sparse system of equations can be efficiently solved by some Krylov iterative solver.

4.2.3 Advantages and Disadvantages

The main advantage of the above method is that it minimizes computational cost by only finding the optimal gain K. The explicit solution to the Algebraic Riccati Equation is never formed. The method is also very stable. In addition, it can easily take advantage of possible sparsity in the control system.

The first disadvantage of the method is that it is only efficient when both $m \ll n$ and $p \ll n$ hold. If we have a large number of controls, or if we wish to control a large number of components of the system, then the above method will loose its advantage.

The second disadvantage comes from the fact that the method will not be easily parallelized. Even though the action of the operator $B\hat{K}$ onto a vector requires only $O(mn)$ number of operations, the matrix $B\hat{K}$ is essentially dense, which may require a large communication overhead.

4.3 Matrix Sign Method

The method described in sections (4.1) and (4.2) looses its advantages when Q has full rank or A is large and dense. In addition it is hard to parallelize. Thus, alternative methods are being developed to exploit modern computing architectures.

The Matrix Sign Method for Riccati equations is a promising alternative. Matrix Sign is an iterative method that requires only a series of linear solves. The linear systems are all symmetric indefinite and there are efficient algorithms that take advantage of the problem structure. A disadvantage of those linear systems is that they are all dense even if the input matrices from the Riccati problem are sparse. The main advantage of the algorithm is that it offers possibilities for efficient paralellization.

4.3.1 Description of the Algorithm

We define the Matrix Sign Function as follows [9]:

Definition 1 ***Matrix Sign Function***

Given matrix Z,

$$sign\,(Z) = \lim_{k\to\infty} Z_k$$

where

$$Z_0 = Z \qquad and \qquad Z_{k+1} = \frac{1}{2}\left(Z_k + Z_k^{-1}\right)$$

In actual computation of the *sign* of a matrix, the iteration

$$Z_{k+1} = \frac{1}{2}\left(Z_k + Z_k^{-1}\right)$$

can be replaced by:

$$Z_{k+1} = \frac{1}{2}\left(\frac{1}{c_k}Z_k + c_k Z_k^{-1}\right),$$

were the sequence c_k is chosen to speed convergence. A good choice is $c_k = det(Z_k)^{\frac{1}{n}}$ where n is the size of Z. Computation of the determinant of a matrix can be expensive, because it requires a factorization, however in practice, the factorization is already available from the computation of Z_k^{-1}. Introduction of c_k can dramatically decrease the number of iterations required for the problem to converge. In test problems c_k decreased the number of iterations by $30 - 40\%$.

The Matrix Sign Function has many properties. The one most useful in solving the Riccati equation is given by the fact that if

$$Z = H \begin{pmatrix} N & T \\ 0 & P \end{pmatrix} H^{-1},$$

where eigenvalues of N have negative real part (stable) and eigenvalues of P have positive real part (unstable), then

$$sign(Z) = H \begin{pmatrix} -I_N & 0 \\ 0 & I_P \end{pmatrix} H^{-1},$$

where I_N is the identity matrix with dimensions corresponding to N, and I_P is the identity matrix with dimensions corresponding to P [9].

The above property can be used to construct a method for finding the solution to the control problem. Given the differential equation

$$\dot{x}(t) = Ax(t) + Bu(t)$$

with some initial condition $x(0) = x_0$. We seek an optimal feedback control that minimizes

$$J(u(\cdot)) = \int_0^\infty \langle x(t), Qx(t)\rangle + \langle u(t), Ru(t)\rangle \, dt,$$

where Q is symmetric positive semi-definite and R is symmetric positive definite. According to the Maximal Principle [5] the optimal control is given by $u(t) = -Kx(t)$, where the optimal gain K is given by $K = R^{-1}B^T\Pi$. If (A, B) is a stabilizable pair [5], then the operator Π is the unique positive semi-definite solution to the Algebraic Riccati Equation

$$A^T\Pi + \Pi A - \Pi B R^{-1} B^T \Pi + Q = 0.$$

The closed loop system is therefore given by

$$\dot{x}(t) = \left(A - BR^{-1}B^T\Pi\right)x(t)$$

where the matrix

$$F = \left(A - BR^{-1}B^T\Pi\right)$$

is stable.

Since F is stable and since the matrix $W = BR^{-1}B^T$ is symmetric positive semi-definite, the Lyapunov Equation

$$FV + VF^T + W = 0 \tag{4.5}$$

has a unique symmetric solution V [10]. If n is the size of the control problem (i.e. the size of A), we can define the $2n \times 2n$ matrix U as

$$U = \begin{pmatrix} A & -W \\ -Q & -A^T \end{pmatrix}. \tag{4.6}$$

Using the solution to the Lyapunov equation, U can be factorized as

$$U = \begin{pmatrix} I & -V \\ \Pi & I - \Pi V \end{pmatrix} \begin{pmatrix} F & 0 \\ 0 & -F^T \end{pmatrix} \begin{pmatrix} I - V\Pi & V \\ -\Pi & I \end{pmatrix}.$$

We can observe that

$$\begin{pmatrix} I & -V \\ \Pi & I - \Pi V \end{pmatrix} \begin{pmatrix} I - V\Pi & V \\ -\Pi & I \end{pmatrix} = I_{2n}.$$

Then we can apply the properties of the Matrix Sign Function to obtain

$$sign(U) = \begin{pmatrix} I & -V \\ \Pi & I - \Pi V \end{pmatrix} \begin{pmatrix} -I & 0 \\ 0 & I \end{pmatrix} \begin{pmatrix} I - V\Pi & V \\ -\Pi & I \end{pmatrix}$$

and

$$\frac{1}{2}\left(I_{2n} + sign(U)\right) = \begin{pmatrix} V\Pi & -V \\ -(I - \Pi V)\Pi & I - \Pi V \end{pmatrix}.$$

Note that the two columns of the system differ by $-\Pi$. Thus if we represent

$$\frac{1}{2}\left(I_{2n} + sign(U)\right) = \begin{pmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{pmatrix},$$

we can recover Π by solving

$$\begin{pmatrix} S_{12} \\ S_{22} \end{pmatrix} \Pi = -\begin{pmatrix} S_{11} \\ S_{21} \end{pmatrix}$$

It can be shown that if (A, B) is stabilizable, then the above system has a unique solution [7, 9].

From a computational point of view, U is a non-symmetric dense matrix and computation of its inverse can be very expensive. While we cannot change the density of U an alternative iteration that involves only symmetric matrices can be devised. Let $J = (-I_{2n})^{\frac{1}{2}}$. Then

$$J = \left(\begin{array}{cc} 0 & I_n \\ -I_n & 0 \end{array} \right) \text{ and } JU = \left(\begin{array}{cc} -Q & -A^T \\ -A & W \end{array} \right). \tag{4.7}$$

Thus $-J = J^{-1} = J^T$ and JU is a symmetric matrix. If $U_0 = U$ and U_k are the iterates from the matrix sign iteration, then we know that

$$U_{k+1} = \frac{1}{2}\left(U_k + U_k^{-1}\right).$$

If we multiply each step by J we obtain

$$JU_{k+1} = \frac{1}{2}\left(JU_k + JU_k^{-1}\right) = \frac{1}{2}\left(JU_k + JU_k^{-1}J^{-1}J\right).$$

If $Z_k = JU_k$ and if $Z_0 = JU_0$, then

$$Z_{k+1} = \frac{1}{2}\left(Z_k + JZ_k^{-1}J\right) = \frac{1}{2}\left(Z_k - J^T Z_k^{-1}J\right).$$

Z_0 is symmetric, then if Z_k is symmetric so is Z_k^{-1} and so is $J^T Z_k^{-1} J$. Since the sum of two symmetric matrices is symmetric, Z_{k+1} is symmetric. Therefore by induction all the iterates Z_k are symmetric an they converge to $Jsign(U)$. To speed convergence we can use $c_k = det(Z_k)^{\frac{1}{2n}}$.

Finally the Matrix Sign Method for solution of the Riccati Equation is given by the following:

Set

$$U = \left(\begin{array}{cc} A & -W \\ -Q & -A^T \end{array} \right).$$

Let $Z_0 = JU$ and iterate

$$Z_{k+1} = \frac{1}{2}\left(\frac{1}{c_k} Z_k - c_k J^T Z_k^{-1} J\right),$$

where $c_k = det(Z_k)^{\frac{1}{2n}}$. We iterate until some desired convergence tolerance is reached. Then if

$$Z = \left(\begin{array}{cc} Z_{11} & Z_{12} \\ Z_{21} & Z_{22} \end{array} \right)$$

the solution to the Riccati Equation is given by

$$\begin{pmatrix} Z_{22} \\ Z_{12} + I_n \end{pmatrix} \Pi = \begin{pmatrix} I_n - Z_{21} \\ -Z_{11} \end{pmatrix},$$

which can be found in the least square sense by QR factorization.

For finding the limit of the iteration Z_k, we need a good way for solving symmetric indefinite systems. Since at each step the right hand side of the system (i.e. J) is of size $2n$, we would wish to factorize Z_k and then use the factorization to solve for all the columns of $Z_k^{-1}J$.

4.3.2 Linear Symmetric Indefinite Solver

Given an indefinite matrix A, we wish to solve the system $Ax = b$ via some factorization of A. One of the most basic matrix factorizations is $A = LU$, where L is lower triangular and U is upper triangular matrix. In practice this factorization is useful for only a number of special cases, such as when A is diagonally dominant. In general the $A = LU$ factorization is unstable. To make it stable, we use $A = P^T LU$ where P is a permutation matrix. In a special case, however, when A is symmetric, there exists the factorization $A = LDL^T$, where D is a diagonal matrix with the eigenvalues of A and L is lower triangular. The advantage of this factorization is that only L needs to be computed using the lower half of A and thus $A = LDL^T$ requires only half the work of $A = LU$. A disadvantage is that we cannot simply permute A by some matrix P, because the new matrix PA may not be symmetric. Thus when permuting, we need to use symmetric permutations PAP^T. Therefore the permuted symmetric factorization is given by:

$$A = P^T LDL^T P$$

For the case when A is definite we can easily take the square root of D and have $A = P^T \hat{L}\hat{L}^T P$ for the positive case and $A = -P^T \hat{L}\hat{L}^T P$ for the negative case. The above is called the Cholesky factorization and the algorithm is numerically stable for positive and negative definite matrices.

The case where the matrix A is indefinite, the factorization $A = P^T LDL^T P$ is unstable. It is easy to see using

$$A = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}. \tag{4.8}$$

For this matrix the factorization $A = P^T LDL^T P$ does not exist. A is invertible (in fact $A^{-1} = A$), however, the eigenvalues of A are $-1, 1$. Thus

$A = P^T LDL^T P$ cannot be used for general symmetric indefinite matrices. The solution to this problem was proposed by Bunch and Kaufman in 1976 [3]. In the Bunch Kaufman algorithm the factorization $A = P^T L\hat{D}L^T P$, where $\hat{D}$ is block diagonal matrix consisting of 1×1 and 2×2 blocks. It can be shown that the above factorization is stable and the 2×2 blocks of D correspond to pairs of positive/negative eigenvalues of A.

The basic idea of the Bunch-Kaufman algorithm faces the problems with pivoting at each step. In order to preserve symmetry of the matrix A and to be able to work only on its lower half, we need to use symmetric permutation PAP^T. In other words, whenever we permute the i, j rows of A, we need to also permute the i, j columns. Thus the symmetry is preserved; however, this strategy has some limitations. Every element that is originally on the diagonal of A, after the permutation must stay on the diagonal and if it is not on the diagonal it cannot move to the diagonal. Thus it may not be possible to select a stable pivot from the diagonal alone. This is illustrated by the example (4.8), all diagonal entries are 0 and thus they cannot be used for pivots. Bunch-Kaufman gives a pivoting strategy which at each step selects a stable pivot, by not only looking at the diagonal entries but also at the off-diagonal entries. In case the first off-diagonal element at step i is sufficiently bigger than the diagonal element, the 2×2 block can be used as a stable pivot. For the inverse of 2×2 matrix there is very easy explicit formula.

The Bunch-Kaufman pivoting strategy is that at each step a decision about the pivot is made based on the following rules. In the elimination of column j check for the following cases:

D1: $|A_{jj}| \geq \alpha |A_{ij}|$, where $j < i < N, |A_{ij}| = max_{k=j+1}^{N} |A_{kj}|$

D2: the conditions D1 and D4 do not hold, use A_{jj}

D3: the 1×1 pivot from A_{ii} will be stable

D4: the 2×2 pivot from columns i, j will be stable

For the case of D1 no interchange is required. The optimal value of α has been computed by Bunch in his original paper and $\alpha = \frac{1+\sqrt{17}}{8}$. Note that in actual experiments for matrices exhibiting our structure (4.7), we found that when α was too far from α_{opt}, the solver was unstable. For the case D3, an interchange between row/columns i, j is required and the newly obtained 1×1 pivot will be stable. In the D4 case if no stable pivot can be obtained from the diagonal, a 2×2 block will be formed from the diagonal and the first off-diagonal element. Two symmetric interchanges will have to be performed, however, both $j, j+1$ columns will be eliminated. For the 2×2 pivot to be stable, if the pivot is represented as:

$$D = \begin{pmatrix} d_1 & c \\ c & d_2 \end{pmatrix}$$

the stability conditions are explicitly $c > \alpha \; max(d_1, d_2)$. Thus $det(D) < 0$, the pivot leads to stable factorization and is an indefinite diagonal block. The condition D2 can lead to an unbounded L and modified Bunch-Kaufman algorithm has been developed. However, the stability of the standard Bunch-Kaufman algorithm is sufficient for most application and is the version we implemented for our Matrix Sign Algorithm.

The factorization algorithm is described in the graph below:

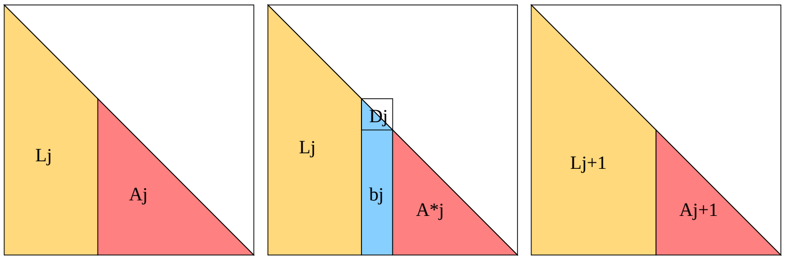


Figure 4.1: Illustration of Bunch-Kaufman Algorithm

Given the lower triangular part of the matrix A, we eliminate the columns of A and over write them with the columns of L. At step j the $1 \ldots j-1$ columns of A have been eliminated: L_j represents the first j columns of L and A_j represents the remaining part of A that has yet to be factorized. A pivot D_j is chosen via the Bunch-Kaufman pivoting strategy and the necessary interchanges of the A_j and L_j are performed. All the interchanges are stored in a permutation vector P (not shown on figure). Then the block b below the pivot D_j is updated by $l = bD_j^{-1}$. The remaining part A_j^* is updated to $A_j^* = A_j^* - lD_j l^T$. Depending on whether a 1×1 or 2×2 block was used on the next step L_{j+1} or L_{j+2} is L_j concatenated with l and A_{j+1} or A_{j+2} is A_j^*. The pivots D_j are stored in a separate vector D. When the algorithm terminates the lower part of A is now L and we have the permutation vector P and the block diagonal D.

The next stage of the Matrix Sign Algorithm is the inversion of A. We are interested in computing $J^T A^{-1} J$. Using the Bunch-Kaufman Algorithm, we have already computed the factorization $A = P^T LDL^T P$. Thus:

$$J^T A^{-1} J = J^T P^T L^{-T} D^{-1} L^{-1} PJ = S^T D^{-1} S, \tag{4.9}$$

where $S = L^{-1}PJ$. During this stage, it is enough to compute only S and then compute the product. This gives much better performance than explicitly computing L^{-T}. The main advantage is that J is very sparse, it has only one non-zero element per column/row and P is only a permutation of the rows. This sparsity can be used in the implementation of $LS = PJ$. The lower-triangular solve fills all the entries of a vector below the first non-zero entry. Therefore, the solve requires only about half of the work needed for a dense solve. S is a lower triangular matrix with permuted columns. Since D is block diagonal with 1×1 and 2×2 blocks the inverse can be computed explicitly and efficiently. If we solve directly for L^T the resulting matrix will be full despite the partial sparsity of S. If we compute $S^T D^{-1} S$, the sparsity of S can be used efficiently. One of the best implementations of Bunch-Kaufman on a single processor is given by LAPACK, their code, however, does not take advantage of potential sparsity of the right hand side vectors. Our code written to take advantage of the sparsity can outperform LAPACK by more than a factor of two.

Given the factorization $A = P^T LDL^T P$, we note that $det(P) = \pm 1$ and L is lower triangular with ones on the main diagonal $det(L) = 1$. Thus:

$$|\det(A)| = \left|\det(P^T)\det(L^T)\det(D)\det(L)\det(P)\right| = |\det(D)|\,.$$

Since D is block diagonal with the blocks denoted by D_j, we can compute

$$|det(A)| = |\det(D)| = \Pi_{j=1}^{k} |\det(D_j)|\,.$$

For large problems this number can easily exceed machine precision, however, we only need to find $c = |det(D)|^{\frac{1}{N}}$. This the product can be computed in several parts and the power can be applied on every sub product separately.

At the end of each step, having computed $c = |det(D)|^{\frac{1}{N}}$ and $B = J^T A^{-1} J$, the next iterate i simply given by:

$$A \leftarrow \frac{1}{2}\left(\frac{1}{c}A - cB\right)$$

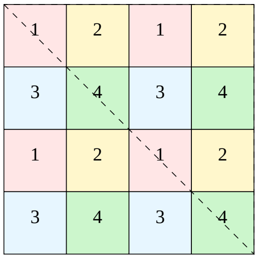
Note that the number of operations required to do each of the three stages of an iteration is of order $O(n^3)$. On a single processor, however, the distribution of compute time is approximately 2 : 3 : 1. Thus the solve for L stage is by far the most expensive one.

4.3.3 Parallel Implementation

A parallel implementation of the Bunch-Kaufman algorithm for distributed memory architectures is not included in any standard parallel package. We implemented the Bunch-Kaufman algorithm specifically in the case of the Matrix Sign Method. For main reference we used [13]. This reference provides a description of the parallel algorithm. However, the authors of [13] consider only the case when we have a few right hand sides. Since in our case we have to solve for all $2n$ columns of J we had to modify the algorithm. As shown in the single processor example, the solve stage of the algorithm requires the largest amount of work, thus sparsity of the right hand side had to be used to its fullest potential. Furthermore the trade off between the increased time for factorization versus the decreased time for the solve is in general justified.

The most expensive part in the factorization of the matrix is the number of symmetric exchanges that have to be made. For every row interchange, a column interchange has to be performed, so it seems that twice the work of an LU interchange has to be performed. However, since A is symmetric and since we are working only on the lower half of A, the amount of information to be exchanged is the same as LU. The amount of computational work needed, however, is half of that of LU, so the ratio of communication to computation in Bunch-Kaufman algorithm is twice that of LU. Thus the communication becomes the bottleneck of the factorization.

The optimal way of distributing memory in linear algebra is to use a logical grid of processors. Thus the matrix A will be split into blocks and distributed among processors on the grid. The cyclic scheme used by [13] is optimal for the factorization stage When a large number of right hand sides have to be solved, it is better to use a scheme with larger blocks. In the current implementation, on a processor grid of 4 processors, a matrix A will be distributed as shown in Figure 4.2. A will be located below the diagonal line.



1	2	1	2
3	4	3	4
1	2	1	2
3	4	3	4

Figure 4.2: Illustration of Bunch-Kaufman Algorithm

In the implementation of the algorithm, special attention has to be paid to the triangular solve stage. In its essence the tri-diagonal solve is a sequential algorithm and cannot be parallelized. However, it can be pipelined. Thus at each moment several right hand sides are in different stages of the solve. Processors on the diagonal will solve for the next entries of some vectors, while processors off the main diagonal will form local sums needed by the diagonal processors to form the next entries of other vectors.

The last stage of the solve is of the form of $B = S^T D^{-1} S$. A matrix inner product like this one is very expensive due to communication cost. Thus full use of partial sparsity has to be made. Because of the enormous communication cost, this last part takes longer than either the tri-diagonal solve or the factorization. Only D^{-1} can be computed in parallel with very little cost, because of the block diagonal structure of D.

The parallel implementation of the above algorithm was done on UNIX platforms, in particular Linux and OSX, using the GNU gcc C compiler. The systems were parallel machines build on the cluster model, using distributed memory and MPI for the inter processor communication procedures.

4.3.4 Numerical Results for Matrix Sign Method

The first problem is of size 1000. Thus, at each step we have to factorize a matrix of size 2000×2000 and solve for 2000 right hand side equations. The Matrix Sign took 15 iterations to converge to desired tolerance $1.e-12$. The time per iteration includes the factorization, solve and forming $S^T D^{-1} S$, as described in the implementation section. On two processors the solver took $53 sec$ per iteration. The total run time was $20 min$ including the time to load the data across the processor cluster and solve the final least square problem using QR factorization.

The second problem has size 2000 and requires the solution of systems of equations of size 4000. The method converged in approximately 17 iterations to the desired tolerance. Below is a table with the time per iteration required for 4, 9 and 16 processors (see Table 4.1).

An algorithm is perfectly parallel if $t \times p = const$, where t is the execution time and p is the number of processors. Strazdins [13] argues that perfect paralellization is impossible for the Bunch-Kaufman algorithm. In our case the task is even more complicated because of the large number of right hand sides.

For a problem of size 3000 we obtain the results in Table 4.2.

Based on the above results, we can make the following observations. The first observation is based on the improvement of time versus the number of processors used. The relative difference between a hypothetical perfect

Table 4.1: Benchmarks for Matrix Sign Method - size 2000

number of processors	time
4	$4.7min$
9	$3.1min$
16	$2.3min$

Table 4.2: Benchmarks for Matrix Sign Method - size 3000

number of processors	time
4	$15.3min$
9	$8.6min$

implementation and our implementation decreases. The second is that as the problem size increases the amount of work increases of order $O(n^3)$, as expected.

For a problem with size 5000 on 9 processors, the program requires $37.8min$ per iteration. For a problem with size 8000 on 9 processors, the time per iteration is $2.48h$. Due to technical difficulties, more than 9 processors could not be used.

The above experiments were simply testing the solver. A control problem with size 3882 converged using 6 processors in 7.5 hours. If the same problem is loaded in Matlab, the LQR command gives an error that there is not enough memory on a machine with $1GB$ RAM. The problem can be solved with the Chandrasekhar algorithm on a single machine in 3 weeks. The 3882 problem was solved with a less efficient implementation of the algorithm. Currently a better implementation is under construction. The new implementation will be able to solve the problem in approximately $5.5h$, however, it is not yet stable enough to generate meaningful results.

All the above examples were run on a Beowulf Linux cluster. The Virginia Tech SystemX supercomputer give approximately 30% better performance.

4.4 Iterative Refinement

The methods described above are ways to create numerical approximations to the solution to a Riccati Equation and all numerical methods are imperfect.

Given a numerical method, there is always a problem for which the method will fail to converge. In order to achieve optimal results, we sometimes have to use combinations of methods.

The Matrix Sign Method is the most unstable method, among the methods we discussed so far. In order for the Matrix Sign to be backwards stable, it needs to be coupled with iterative refinement [7]. If we wish to solve the ARE

$$0 = A^T\Pi + \Pi A - \Pi BR^{-1}B^T\Pi + Q,$$

we can use Matrix Sign Method to obtain an approximate $\hat{\Pi}$. We can define $\Delta\Pi$ as $\Pi = \hat{\Pi} + \Delta\Pi$. We let the residual H_{Res} be

$$H_{Res} = A^T\hat{\Pi} + \hat{\Pi}A - \hat{\Pi}BR^{-1}B^T\hat{\Pi} + Q.$$

Then, we can derive the equation for $\Delta\Pi$ as

$$0 = \left(A - BR^{-1}B^T\hat{\Pi}\right)^T \Delta\Pi + \Delta\Pi\left(A - BR^{-1}B^T\hat{\Pi}\right) - \Delta\Pi BR^{-1}B^T\Delta\Pi + H_{Res}.$$

The new equation is another Riccati Equation. The key assumption is that if $\hat{\Pi}$ is close to Π, then $\left(A - BR^{-1}B^T\hat{\Pi}\right)$ will be stable matrix and the residual ARE will be better conditioned and thus easier to solve. Gardiner [7] says:

> A common misconception is that it does not matter what method is used to solve the correction equation. In fact the achievable accuracy does depend on the method.

Gardiner describes an example in which the Matrix Sign Method plus Newton iterations give ten more significant digits than the Matrix Sign Method by itself. Given an approximate $\hat{\Pi}$, obtained by the Matrix Sign Method, we can derive the residual Riccati Equation and use the Chandrasekhar Method and/or the Modified Newton Method as described by Banks and Ito.

The Matrix Sign Method is in general the most unstable method, however, Chandrasekhar and Newton can also show instabilities. In a numerical experiment with a very ill-conditioned Riccati problem, the Matlab LQR command gave residual with norm approximately 2.3. The LQR command is based on the Newton Methods and even though the Newton Method is generally considered the best method available, iterative refinement with LQR again, fails to improve the residual. An iterative refinement step with Matrix Sign method gave residual of order $1.e-12$. In order for a good approximate solution to be obtained, we may have to use all three methods together.

Banks and Ito give a good way to use Chandrasekhar solution as initial guess to a Newton iteration. In general, if we wish to find a solution to ARE using

Chandrasekhar alone, we may have to integrate Chandrasekhar to a very large time t, which may give poor accuracy and/or stability. If we partially integrate Chandrasekhar, then we want to use the approximate gain K as initial guess of some form for the Matrix Sign Method. There are two ways to do that. The first way is to observe that the residual H_{Res} is approximately $\dot{\Pi}(\hat{t})$ for some $\hat{t}$. From the derivation of the Chandrasekhar equations we know that $\dot{\Pi}(t) = L^T(t)L(t)$, therefore, if we integrate Chandrasekhar until some time t^* and if $\hat{K} \approx K(t^*)$ and $L \approx L(t^*)$, we can solve the following ARE

$$0 = \left(A - B\hat{K}\right)^T \Delta\Pi + \Delta\Pi \left(A - B\hat{K}\right) - \Delta\Pi BR^{-1}B^T\Delta\Pi + L^T L.$$

Then the optimal gain for the original ARE problem will be given by $K = \hat{K} + R^{-1}B^T\Delta\Pi$. This method is good for obtaining the remainder of the solution to the Chandrasekhar integration, however, it will carry over any numerical error accumulated in the integration until t^*. In order to achieve better accuracy, a higher order method could be used in the Chandrasekhar integration, however, it will decrease stability.

Chandrasekhar and modified Newton methods both give only an approximation to the optimal gain K. Matrix Sign Method computes Π, which is the solution to an ARE, therefore, given a guess $\hat{K}$ we need a way to approximate $\hat{\Pi}$ so that we can derive a regular refinement equation and use the Matrix Sign Method. If we are given $\hat{K}$ and if $\left(A - B\hat{K}\right)$ is stable, then $\hat{\Pi}$ is given as the solution to the Lyapunov Equation [1, 10]

$$\left(A - B\hat{K}\right)^T \hat{\Pi} + \hat{\Pi}\left(A - B\hat{K}\right) + \hat{K}^T RK + Q = 0.$$

The solution to the above equation can be computationally expensive, but it will give us a good way to approximate $\hat{\Pi}$ and thus use Matrix Sign Method as refinement to both Chandrasekhar and Modified Newton Method.

The Chandrasekhar Method can also be used as refinement method to either Newton or Matrix Sign. Given a Riccati equation of the form

$$0 = \left(A - BR^{-1}B^T\hat{\Pi}\right)^T \Delta\Pi + \Delta\Pi\left(A - BR^{-1}B^T\hat{\Pi}\right) - \Delta\Pi BR^{-1}B^T\Delta\Pi + H_{Res},$$

we wish to factor $H_{Res} = C^T C$ and solve Chandrasekhar equations with $\Delta K(0) = 0$ and $L(0) = C$. The Chandrasekhar Method is most efficient, when C is a small matrix (i.e. rank of H_{Res} is small). Numerically H_{Res} can have full rank, however, we may only consider the eigenvalues of H_{Res} that are bigger than some tolerance. If most of the eigenvalues of H_{Res} are small, we can use some form of SVD factorization to ignore those eigenvalues and obtain a small C.

Practical application sometimes result in ill-conditioned ARE equations. In that case, we have to use a combination of methods to find a good approximation to the optimal K. Chandrasekhar, Newton and Matrix Sign Methods can be used together to achieve better stability and accuracy.

Chapter 5

Numerical Results

5.1 Simple 5×5 Problem

We consider the simple 5×5 DAE

$$\left(\begin{array}{ccc|cc} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{array}\right) \begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} = \left(\begin{array}{rrr|rr} 2 & -1 & 0 & 1 & -2 \\ -1 & 2 & -1 & -1 & 2 \\ 0 & -1 & 2 & 1 & 2 \\ \hline 1 & -1 & 1 & 0 & 0 \\ -2 & 2 & 2 & 0 & 0 \end{array}\right) \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \left(\begin{array}{c} 1 \\ 0 \\ 0 \\ \hline 0 \\ 0 \end{array}\right) u.$$

We are interested in minimizing the cost

$$J(u(\cdot)) = \int_0^\infty \langle x_1(t), I_3 x_1(t) \rangle + \langle u(t), 1u(t) \rangle \, dt.$$

We take the initial condition to be $x_1^0 = (1,1,0)^T$. The DAE system is unstable, if we solve numerically using the Backward-Euler method with $\Delta t = \frac{1}{1024}$, from 0 until $t = 20$, we have that $\|x(20)\| = O(10^9)$ and $J(0) = O(10^{17})$.

We want to find the optimal gain K so that $u(t) = -Kx_1(t)$ is the optimal control that will stabilize the system and minimize $J(\cdot)$. Next we apply the three methods discussed in Chapter 3.

Using the Matlab SVD command we obtain

$$V = \begin{pmatrix} -0.7071067811 \\ -0.7071067811 \\ 0 \end{pmatrix} \qquad \bar{V} = \begin{pmatrix} -0.6139366992 & 0.3508300576 \\ 0.6139366992 & -0.3508300576 \\ 0.4961486256 & 0.8682376064 \end{pmatrix}.$$

With V and $\bar{V}$ we can solve the Riccati DAE using Backward-Euler method with time step $\Delta t = \frac{1}{1024}$ until the difference between two successive iterates is less than 1.e-8. Newton's method was used at each step to solve the nonlinear equation. The solution obtained is

$$\Pi = \begin{pmatrix} 2.22474382870834 & 2.22474382870834 & 0 \\ 2.22474382870834 & 2.22474382870834 & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

The minimum norm solution in this case is symmetric positive definite. The optimal gain is

$$K = \begin{pmatrix} 2.22474382870834 & 2.22474382870834 & 0 \end{pmatrix}.$$

If we simulate the DAE with control $u(t) = -Kx_1(t)$, we obtain that

$$\|x(20)\| = 9.1515\text{e-}011 \qquad \text{and} \qquad J(u(t)) = 8.89366090338702.$$

The feedback control stabilized the system and gave small finite cost.

Next we try to obtain the optimal gain by solving the Riccati equation for the Change of Variable system. The system is a 1×1 differential system and using Matlab LQR command we find the Π for the DAE system to be

$$\Pi_v = \begin{pmatrix} 2.22474487139159 & 2.22474487139159 & 0 \\ 2.22474487139159 & 2.22474487139159 & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

The Π from Riccati DAE and the Change of Variable agree in the first six significant digits. The error is $\|\Pi - \Pi_v\| = 4.1707\text{e-}006$. The error in the optimal gain is $\|K - K_v\| = 2.0854\text{e-}006$. Simulating the DAE with $u(t) = -K_v x_1(t)$ gives $\|x(20)\| = 9.1513\text{e-}011$ and $J(u(t)) = 8.89366089886046$.

The difference between the two methods is in the range of round-off error. This confirms the theoretical prediction made in Chapter 3.

Next we wish to apply the penalty method. We use $M = I_3$. Below is a table with values for the difference between $\Pi(\epsilon)$ and the optimal gain given by the Change of Variable method (see Table 5.1).

Convergence of $\Pi(\epsilon)$ to Π_v appears to be linear. It is interesting to note that after $\epsilon = 1.\text{e-}3$, the improvement on $\|x(20)\|$ and $J(\cdot)$ seems to be negligible. For this example $\Pi(\epsilon)$ converges to the minimum norm solution Π.

Table 5.1: Convergence of the Perturbation Method - Symmetric Π

ϵ	$\|\Pi(\epsilon) - \Pi_v\|$	$\|x(20)\|$	$J(u(\cdot))$
1.e-2	5.6500e-002	8.3581e-011	8.89367515544928
1.e-3	5.6000e-003	9.0694e-011	8.89365926322830
1.e-4	5.5885e-004	9.1431e-011	8.89366070511796
1.e-5	5.5880e-005	9.1505e-011	8.89366087918493

Table 5.2: Convergence of the Perturbation Method - Non-Symmetric Π

ϵ	$\|\Pi(\epsilon) - \Pi\|$	$\|\Pi_v(\epsilon) - \Pi\|$	$\|x(20)\|$	$J(u(\cdot))$
1.e-2	4.4848296022	7.05532e-002	2.7015e-007	13.3646229193
1.e-3	4.4529856446	6.98470e-003	2.9171e-007	13.3644460035
1.e-4	4.4498389566	7.03365e-004	2.9394e-007	13.3644374951
1.e-5	4.4495246601	7.59718e-005	2.9416e-007	13.3644367346

A more interesting example is when we change the system to

$$\left(\begin{array}{ccc|cc} 2 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{array}\right) \begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} = \left(\begin{array}{rrr|rr} 2 & -1 & 0 & 1 & -2 \\ -1 & 2 & -1 & -1 & 2 \\ 0 & -1 & 2 & 1 & 2 \\ \hline 1 & -1 & 1 & 0 & 0 \\ -2 & 2 & 2 & 0 & 0 \end{array}\right) \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \left(\begin{array}{c} 1 \\ 0 \\ 0 \\ \hline 0 \\ 0 \end{array}\right) u.$$

The structure is still consistent, but E is not simply the identity. The optimal Π given by both the Riccati DAE and the Change of Variable is

$$\Pi = \begin{pmatrix} 4.44948765257459 & 4.44948765257459 & 0 \\ 2.22474382628730 & 2.22474382628729 & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

As predicted by the theory, the optimal Π is not symmetric. The norm of the solution at $t = 20$ and the cost until $t = 20$ are

$$\|x(20)\| = 2.9418\text{e-}007 \qquad \text{and} \qquad J(u(t)) = 13.3644.$$

If we apply the penalty method, $\Pi(\epsilon)$ cannot possibly converge to Π, because $\Pi(\epsilon)$ converges to a symmetric solution and Π is not symmetric. Using Lemma 8, we can obtain a minimum norm solution $\Pi_v(\epsilon)$. Table 5.2 describes the numerical experiments.

Table 5.3: Convergence of the Perturbation Method - Heat Equation

N	ϵ =1e-1	ϵ =1e-2	ϵ =1e-3	ϵ =1e-4	ϵ =1e-5	ϵ =1e-6
4	6.5027e-3	6.5993e-4	6.6117e-5	6.6130e-6	6.6131e-7	6.6131e-8
8	6.5025e-3	6.5991e-4	6.6115e-5	6.6128e-6	6.6129e-7	6.6130e-8
16	6.5024e-3	6.5990e-4	6.6115e-5	6.6128e-6	6.6129e-7	6.6129e-8
32	6.5024e-3	6.5990e-4	6.6115e-5	6.6127e-6	6.6129e-7	6.6129e-8
64	6.5024e-3	6.5990e-4	6.6115e-5	6.6127e-6	6.6129e-7	6.6128e-8
128	6.5024e-3	6.5990e-4	6.6115e-5	6.6128e-6	6.6129e-7	6.6141e-8

5.2 Heat Equation

The next numerical experiment is done with the one dimensional heat equation

$$v_t(t,x) = \mu v_{xx}(t,x) + b(x)u(t).$$

The boundary conditions of the heat equation can be written as the algebraic constraint

$$v(t,0) = 0 = v(t,1).$$

We can consider discretized version of the equation with finite element method that has the DAE form discussed in Chapters 2 and 3. We can also discretize the heat equation by imposing the boundary conditions on the finite element basis and obtain a purely differential system. Therefore, we have two ways to approximate the optimal gain K, one is using the well developed theory of purely differential equations and another is using our theory of DAE systems. We apply the penalty method and observe the convergence rate in Table 5.3.

We can see that the gains obtained from the penalty method approximation converges to the optimal gain approximately linearly. It is interesting to note that the convergence seems to be independent from the mesh size. This may be only a property of the heat equation or perhaps a property of the penalty approximation in general.

5.3 Stokes Flow

We consider the two dimensional incompressible fluid flow over a cavity as shown in the picture below (Fig 5.1). The blue nodes represent the inflow and outflow boundary, where the fluid enters and exits the domain. The red nodes represent non-slip boundary, the fluid is at rest at the boundary. The green nodes represent the part of the boundary that we can control. We can control the normal tension to the fluid. Our goal is to minimize a quadratic functional cost and Q puts weight only on the velocity of the fluid inside the cavity.

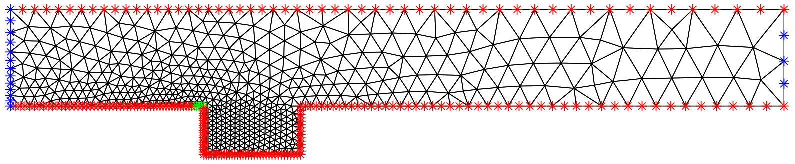


Figure 5.1: The Mesh for Flow over Cavity

We apply the penalty method with $\epsilon = 1.e - 4$. The resulting problem has size 3882. The gain was computed using the matrix sign method on a parallel machine using 6 nodes. The gain in x and y direction is shown in figures (5.2) and (5.3).

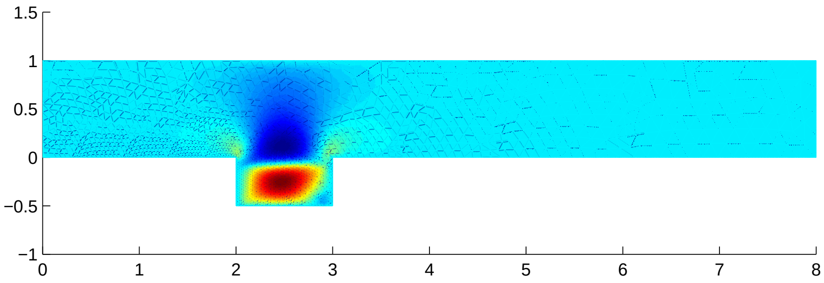


Figure 5.2: The Computed Gain in X Direction

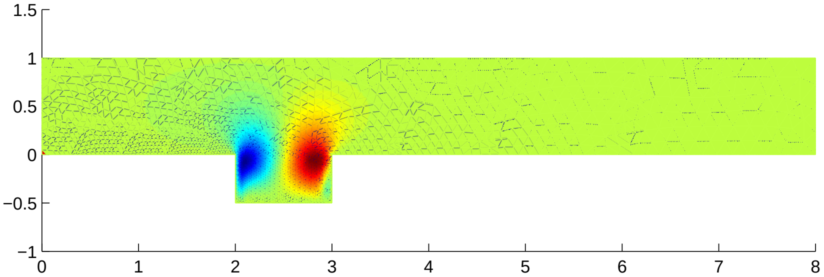


Figure 5.3: The Computed Gain in Y Direction

Chapter 6

Conclusions

We considered optimal linear-quadratic feedback control for differential algebraic equations that come from the discretizations of saddle point problems. We derived necessary conditions for the optimal control and Riccati equations for the optimal gain. Since the Riccati equations were impractical, we considered two alternative ways of finding the optimal gain. The first way is approximating the gain via a penalty method and the second way is performing a change of variable to obtain a purely differential system. Both methods convert the impractical Riccati equation to a standard Riccati equation. We considered ways for solving large scale sparse regular Riccati equations based on the Chandrasekhar and matrix sign methods. We gave numerical examples.

Future work will consists of further exploration of the properties of the DAE systems and the approximate ways for finding the optimal control. Here are some of the questions that we hope to answer in the future:

- We showed that the minimum norm Π does not have to be symmetric, however, in the examples we considered, it was always positive semi-definite. We wish to show that either Π is always positive semi-definite or find an example where that fails.
- Numerical examples suggest that the approximate optimal gains computed by the penalty method converge linearly. We wish to prove that property analytically.
- The Change of Variable method has better potential than the penalty method, because it directly computes the minimum norm optimal gain. The main problem of the Change of Variable is that the basis for $ker(A_{21})$ has to be computed. We can compute the basis using SVD or

QR factorizations, both of which are very computationally expensive. We wish to explore alternatives to SVD and QR. In addition we wish to explore potential ways to preserve sparsity.

- The matrix sign method for standard Riccati equations is an attractive method because it can be easily parallelized. The main problem of the method is its stability. We wish to explore ways to efficiently combine matrix sign and the Newton method and thus improve stability.

- We wish to run more numerical experiments involving fluid flow. We wish to consider larger and more complicated domains as well as denser meshes.

Bibliography

[1] H. T. Banks and K. Ito. A numerical algorithm for optimal feedback gain in high dimensional linear quadratic regulator problems. *SIAM J. Control and Optimization*, vol.29 No.3, May 1991.

[2] Douglas J. Bender and Alan J. Laub. Linear quadratic optimal regulator for descriptor systems. *IEEE Transactions on Automatic Control.*, vol.32 No.8, August 1987.

[3] James R. Bunch, Linda Kaufman, and Beresford N. Parlett. Decomposition of a symetric matrix. *Numerische Mathematik*, vol.27, 1976.

[4] S. L. Campbell. *Singular Systems of Differential Equations.* Pitman Publishing Limited, 1980.

[5] John. L. Casti. *Dynamical Systems and Their Applications.* Academic Press, Inc., 1977.

[6] Daniel Cobb. Descriptor variable systems and optimal state regulation. *IEEE Transactions on Automatic Control.*, vol.28 No.5, May 1983.

[7] Judith D. Gardiner. A stabilized matrix sign function algorithm for solving algebraic Riccati equations. *SIAM J. Scientific Computing*, vol.18 No.5, September 1997.

[8] R. Glowinski. *Numerical Methods for Nonlinear Variational Problems.* Springer-Verlag., New York, 1984.

[9] Charles S. Kenney and Alan J. Laub. The matrix sign function. *IEEE Transactions on Automatic Control.*, vol.40 No.8, August 1995.

[10] E. B. Lee and L. Markus. *Foundations of Optimal Control Theory.* Robert E. Krieger Publishing Co., 1986.

[11] David. G. Luenberger. *Optimization by Vector Space Methods.* John Wiley Sons, Inc., 1969.

[12] Halsey Royden. *Real Analysis.* Prentice-Hall, Inc., 1988.

[13] Peter E. Strazdins. A dense complex symmetric indefinite solver for the Fujitsu AP3000. *The Australian National University*, TR-CS-99-01, May 1999.