SJÄLVSTÄNDIGA ARBETEN I MATEMATIK

MATEMATISKA INSTITUTIONEN, STOCKHOLMS UNIVERSITET

Bang-Bang control in Reinforcement Learningand System Identification

av

Alexander Westberg

2023 - K14

MATEMATISKA INSTITUTIONEN, STOCKHOLMS UNIVERSITET, 106 91 STOCKHOLM

Bang-Bang control in Reinforcement Learningand System Identification

Alexander Westberg

Självständigt arbete i matematik 15 högskolepoäng, grundnivå

Handledare: Yishao Zhou

2023

Abstract

From numerical results it is observed that a Bang-Bang controller has competitive performance against a continuous controller on different tasks in Reinforcement learning problems. The performance of a Bang-Bang controller in Reinforcement learning problems is yet not fully understood yet, and is thus and open research question. In this paper we explore this open question and provide an partial explanation with mathematical proof why this phenomena exist. We start by understanding the foundations of control theory and the Bang-bang controller. With an existing mathematical foundation of System Identification we derive errors for approximating dynamical systems. Using the error terms, we prove that the Bang-Bang controller yields a better approximation then a continuous controller in Reinforcement learning problems.

Contents

Acknowledgement

I would like to acknowledge and thank my supervisor, Yishao Zhou, for her guidance and support during this thesis. Her expertise in mathematics and control theory greatly influenced the research direction and quality. I appreciate her time, feedback, and wisdom, which helped improve the ideas and methodologies.

1 Introduction

In the paper [6] the researchers provide numerical results in Reinforcement learning problems that when restricting agents to only extreme controls yields competitive performance compared to continuous controllers. More specifically, they only consider the maximum and minimum of an given controller compared to an controller where all the available controllers are considered. In complex control problems where the controls are of high dimensions it seems unlikely that the optimal controller would only be strictly Bang-Bang controller, which is pointed out by the researchers. But the competitive observed results with the Bang-Bang controller does raise a question why it is observed. The researchers from [6] does give several a hypothesis why this phenomena occurs but without mathematical proof. One of them being that when the dynamical system is unknown the Bang-Bang controller enables better exploration. Therefore a better approximation of the given dynamical system.

The goal of this paper is to prove that under reasonable assumptions the Bang-Bang controller yields a better approximation of an unknown dynamical system. We will extend the ideas from the paper [2] which provide a good mathematical foundation for identifying discrete dynamical systems. With the extension of those ideas we will prove that the Bang-Bang controller approximate any discrete dynamical system better then a Gaussian continuous controller under an learning process.

This paper begins with preliminaries in Linear Algebra. Both control theory and Reinforcement learning heavily rely on this subject. Understanding it, will be a crucial part of our examination of control theory, Reinforcement learning and our analysis of the Bang-Bang controller in System Identification.

In section 3 we examine control theory, specifically about Reachability and Optimal control. Here we will closely study different approaches of finding an optimal controller. In section 4 we examine the basics of Reinforcement Learning and draw a distinction between control theory and Reinforcement Learning. In section 5, we will show how to apply Reinforcement Learning; we derive a method for learning a model from data. Then in 5.2, we show how a Bang-Bang controller improve the error of the approximated model.

1.1 Insights from numerical results

In the *Figure 1* we can see the results of using both the Bang-Bang and Gaussian controller the researchers from [6] observed. We can observe that the result with different algorithms and controllers varies between tasks. So it is not clear that any of the two controllers are better then the other. But since these two controllers are quite different from one another, it raises the question of why the controllers have similar performance. The researches provide different hypothesis of this question. Although they do give many ideas, but they are not mathematically proved.

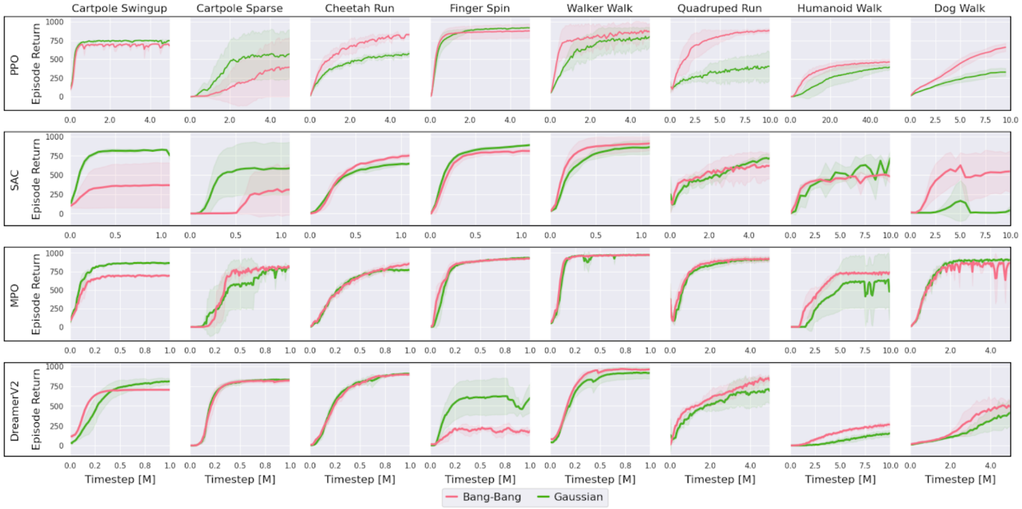

Figure 1: The rows the different algorithms that are being used for the different tasks and the Columns are the different tasks the algorithms are being trained on. Figure taken from [6].

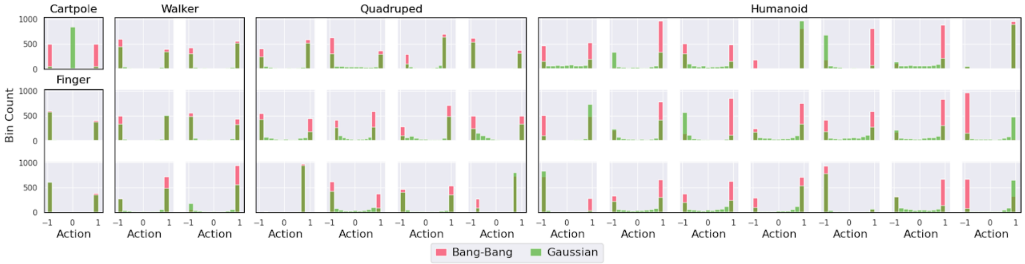

Figure 2: The distributions of magnitude of the controllers for the trajectories of the MPO algorithm in *Figure 1*. Figure taken from [6].

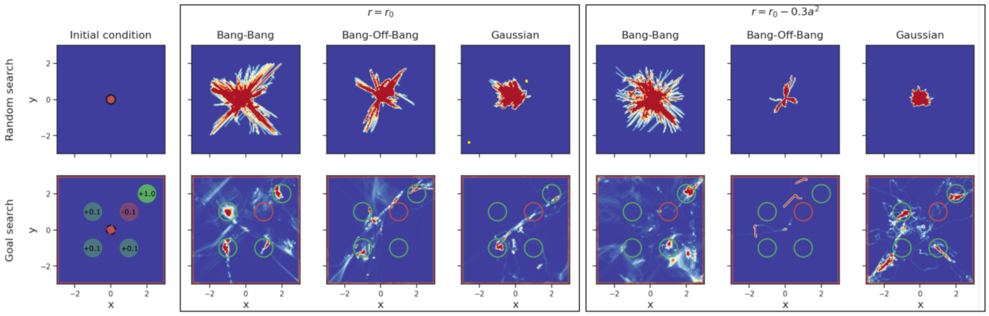

Figure 3: The explored environment by using different kinds of controllers during two tasks. The rows shows the goal of the task and the columns are the different kind of controllers. Figure taken from [6].

In *Figure 2* we can see that even though all the controllers (in an interval) are available to the Gaussian controller, it tend to choose the controllers at the extremes. So, this algorithm (MPO short for Maxmimum A Posteriori Policy Optimization) being used in the RL problems tend converge to an approximate Bang-bang controller.

The goal of this paper is to provide a good mathematical foundation to try to explain these result. We will investigate the theoretical framework of mathematical optimal control. Which deals with deriving an controller given some dynamical system and performance index, rather then learning it from data which is usually done in Reinforcement learning.

Since control theory and Reinforcement learning (RL) both heavily rely on the framework of linear algebra, we will have to build a strong understanding of Linear algebra firstly before we go onto control theory and RL.

1.2 Hypothesis of performance for the Bang-Bang controller

If the environment of an agent is unknown, the agent has to learn its environment. The researchers from [6] give an idea that the using the Bang-Bang controller will result in the agent exploring a larger part of its environment, which is observed in their numerical experiments in *Figure 3*. So when an agent explores a larger area, it will yield better results. The researchers argue that costs of controllers can hinder the Gaussian controller from exploring and finding the optimal controller. While costs of the Bang-bang controller can also hinder the agent from achieving maximum performance due too only choosing maximum action. It is later argued that an optimal design should be a combination of a Bang-Bang controller and Gaussian controller.

But this idea is not proved in that paper. In the *Figure 3* they do show that agents with an Bang-bang controller explores a larger area of its environment.

As said, it is not mathematically shown how a larger area explored will cause a better performance. But it is an idea that we will prove later in this paper.

2 Linear Algebra Preliminaries

Linear algebra plays a pivotal role in the field of control theory, serving as a fundamental mathematical framework for the analysis and design of control systems. By employing mathematical models to represent these systems, we can leverage linear algebra techniques to gain insights into their dynamics and manipulate their responses. The application of linear algebra enables the examination of input-output relationships, investigation of system stability and controllability, design of optimal control strategies.

In this section we will develop propositions, definitions and theorems in linear algebra that will be necessary (directly or indirectly) for our analysis in this paper. Some of the proofs is based on lecture notes and material from [8].

2.1 Definite Matrices

Definite matrices hold a significant importance in control theory. These matrices are square matrices that possess distinct eigenvalue properties, providing crucial insights into system dynamics. Positive definite matrices, in particular, play a central role. Such matrices are associated with stable and well-behaved systems.

Definition 2.1. A symmetric matrix $A \in \mathbb{R}^{n\times n}$ is called a *positive definite* matrix if for all nonzero vectors $x \in \mathbb{R}^n$

$$x^\top Ax > 0.$$

Proposition 2.1. *A positive definite matrix $A \in \mathbb{R}^{n\times n}$ has only positive eigenvalues.*

Proof. If a matrix is *positive definite* then $x^\top Ax > 0$ for all nonzero vectors. Now take any x eigenvector and corresponding eigenvalue λ of A

$$x^\top Ax = x^\top \lambda x = \lambda x^\top x = \lambda||x||_2^2.$$

Since the euclidean norm $||x||_2^2 > 0$ for all nonzero vectors, the eigenvalue λ must the positive since $0 < x^\top Ax = \lambda||x||_2^2$. □

Proposition 2.2. *If $P \in \mathbb{R}^{n\times n}$ is a positive definite matrix then it is invertible.*

Proof. If $Px = 0 = 0 \cdot x$ for some nonzero $x \in \mathbb{R}^n$ then 0 is a eigenvalue of P, which contradicts that P only has only positive eigenvalues. Therefore P must be invertible. □

Proposition 2.3. *Any $A \in \mathbb{R}^{n\times n}$ matrix that only has positive eigenvalue is a positive definite matrix.*

Proof. Now, from the *Spectral theorem,* an invertible symmetric matrix can be decomposed into

$$A = VDV^\top$$

where $V \in \mathbb{R}^{n\times n}$ is an orthonormal matrix where columns are eigenvectors of A and $D \in \mathbb{R}^{n\times n}$ is a diagonal matrix, where the diagonal elements are the eigenvalues of A.

Now take any vector $x \in \mathbb{R}^n$ and some vector $y \in \mathbb{R}^n$ where $y_i \in \mathbb{R}$ are the elements of y

$$x^\top VDV^\top x = y^\top Dy = \sum_{i=1}^{n} \lambda_i \cdot y_i^2 > 0.$$

□

Definition 2.2. A symmetric matrix $A \in \mathbb{R}^{n\times n}$ is called a *positive semi-definite* matrix if for all nonzero vectors $x \in \mathbb{R}^n$

$$x^\top Ax \geq 0.$$

Proposition 2.4. *If $P \in \mathbb{R}^{n\times n}$ is a positive definite matrix and $Q \in \mathbb{R}^{n\times n}$ is a semi-definite matrix then $P + Q$ is positive definite matrix.*

Proof. Since for every nonzero $x \in \mathbb{R}^n$ we have that $x^\top Px > 0$ and $x^\top Qx \geq 0$

$$x^\top (P + Q)x = x^\top Px + x^\top Qx > 0.$$

□

Proposition 2.5. *If $P \in \mathbb{R}^{n\times n}$ is a positive definite matrix then its inverse P^{-1} is also positive definite.*

Proof. For any nonzero $y \in \mathbb{R}^n$ and define $x = Py \in \mathbb{R}^n$. Since P in an invertible matrix the range of Py is $\mathbb{R}^n$. So, we have that

$$x^\top P^{-1}x = y^\top P^\top P^{-1}Py = y^\top Py > 0.$$

□

Proposition 2.6. *If $P \in \mathbb{R}^{n\times n}$ is a positive definite matrix and $A \in \mathbb{R}^{n\times n}$. Then $A^\top PA$ is a positive semi-definite matrix. If A is invertible, then $A^\top PA$ is a positive definite matrix*

Proof. For any nonzero $x \in \mathbb{R}^n$ and some $y \in \mathbb{R}^n$

$$x^\top A^\top PAx = y^\top Py.$$

Since A is not necessarily and invertible matrix, then there exist an x such that $Ax = 0 = y$. Therefore there exist some vector x such that $x^\top A^\top PAx = y^\top Py = 0$.

Now if A is invertible, there does not exist a vector x such that $Ax = 0$, therefore for some nonzero $z \in \mathbb{R}^n$ $Ax = z$ and

$$x^\top A^\top PAx = z^\top Pz > 0.$$

□

Proposition 2.7. *If $A \in \mathbb{R}^{n\times n}$ with dimension $0 < k \leq n$ and singular values $0 < \sigma_1 \leq ... \leq \sigma_k$. Then*

$$\sum_{i=1}^{N} A^i (A^\top)^i$$

is positive semi-definite if $k < n$ and if $k = n$ then it is positive definite, for any N.

Proof. Decompose $AA^\top$ into its Singular Value decomposition, recall that $U, V \in \mathbb{R}^{n\times n}$ are orthonormal matrices, meaning that $UU^\top = U^\top U = I$,

$$AA^\top = (U\Sigma V^\top)(U\Sigma V^\top)^\top = U\Sigma V^\top V\Sigma U^\top = U\Sigma^2 U^\top.$$

Now, since Σ is a diagonal matrix with the singular values of A along its diagonal in a descending order, and the number of singular values depends on the dimension of A.

If $k < n$ then A has k singular values. So, there exists some $y \in \mathbb{R}^n$ such that

$$(U\Sigma^2 U^\top)y = 0$$

since Σ has a dimension of k. We also see that if $U\Sigma^2 U^t x \neq 0$ then

$$x^\top U\Sigma^2 U^\top x = y^\top \Sigma^2 y = \sum_{i=1}^{k} (\sigma_{k-i+1})^2 \; y_i^2 > 0,$$

so it must be positive semi-definite.

Now if $k = n$ then there does not exist $x \in \mathbb{R}^n$ such that $U\Sigma^2 U^t x \neq 0$ meaning that

$$x^\top U\Sigma^2 U^\top x = \sum_{i=1}^{k} (\sigma_{k-i+1})^2 \; y_i^2 > 0$$

for all x and some y. So it must be positive definite.

For the second part of the proof, if $P_1, P_2 \in \mathbb{R}^{n\times n}$ are positive definite then $P_1 + P_2$ is also positive definite, because

$$x^\top (P_1 + P_2)x = x^\top P_1 x + x^\top P_2 x > 0.$$

Therefore if A is invertible then

$$\sum_{i=1}^{N} A^i (A^\top)^i$$

is positive definite.

If $Q \in \mathbb{R}^{n\times n}$ is positive semi-definite then $Q^p + Q^m$ is also positive semi-definite for any $p, m \in \mathbb{R}$, because if $Qx = 0$ then

$$x^\top (Q^p + Q^m)x = x^\top Q^{p-1}(Qx) + x^\top Q^{m-1}(Qx) = 0.$$

Now, if $Qx \neq 0$

$$x^\top (Q^p + Q^m)x = x^\top Q^p x + x^\top Q^m x > 0.$$

Therefore if A is not invertible then

$$\sum_{i=1}^{N} A^i (A^\top)^i$$

is positive semi-definite.

□

2.2 Rayleigh quotient

The Rayleigh quotient is a fundamental concept in linear algebra, providing a means to characterize and analyze the properties of a matrix or a linear operator. It serves as a powerful tool for studying eigenvalues and eigenvectors of a matrix or operator, allowing for the assessment of positive definiteness, and other characteristics. In Control theory, understanding the properties of eigenvalues and eigenvectors is often required to understand certain system dynamics. Therefore knowledge about Rayleigh quotients can be essential for solving control problems. Later in this paper, it will serve as one of our most valuable tools for proving our key results.

Definition 2.3. Given a symmetric matrix $A \in \mathbb{R}^{n\times n}$ and a nonzero vector $x \in \mathbb{R}^n$ we define the *Rayleigh quotient* as

$$R(A,x)\frac{x^\top Ax}{x^\top x}.$$

The following theorem establishes key properties of the Rayleigh quotient, specifically about its maximum and minimum value and its connection to the eigenvectors and eigenvalues of the matrix A. It will be an important tool for later on in the paper.

Theorem 2.1 (Min-max theorem, or Courant–Fischer–Weyl min-max principle)**.** *Let $A \in \mathbb{R}^{n\times n}$ be an symmetric matrix with eigenvalues $\lambda_1 \leq ... \leq \lambda_k \leq ... \leq \lambda_n$. Then*

$$\lambda_k = \min_U\{\max_x\{R(A,x) : x \in U, x \neq 0\} : dim(U) = k\}.$$

In particular

$$\lambda_1 \leq R(A,x) \leq \lambda_n,$$

meaning that

$$\max_x R(A,x) = \lambda_n, \quad \min_x R(A,x) = \lambda_1.$$

Proof. Since A is symmetric matrix, the *Spectral theorem* says that the matrix A is diagonalizable and has an orthonormal basis of eigenvectors $\{u_1, ..., u_n\}$ where u_i is an eigenvector.

Now, if U is a subspace of dimension k, then its intersection with the subspace $span(\{u_k, ..., u_n\})$ is not zero. If it were zero then the span of the two subspaces would be $k + n - k + 1$ which is impossible. Therefore there exists a non-zero vector v in this intersection which we can write as

$$v = \sum_{i=k}^{n} a_i u_i, \quad a_i \in \mathbb{R},$$

and whose Rayleigh quotient is

$$R(A,v) = \frac{(\sum_{i=k}^n a_i u_i)^\top A(\sum_{i=k}^n a_i u_i)}{(\sum_{i=k}^n a_i u_i)^\top(\sum_{i=k}^n a_i u_i)}.$$

We have that $Aa_iu_i = \lambda_i a_i u_i$, so

$$R(A,v) = \frac{(\sum_{i=k}^n a_i u_i)^\top(\sum_{i=k}^n \lambda_i a_i u_i)}{(\sum_{i=k}^n a_i u_i)^\top(\sum_{i=k}^n a_i u_i)}$$

since the vector u_i, u_j are orthonormal vectors $u_i^\top u_j = 0$ and $u_i^\top u_i = 1$

$$R(A,v) = \frac{\sum_{i=k}^n a_i^2 \lambda_i}{\sum_{i=k}^n a_i^2}.$$

Now since $\lambda_1 \leq ... \leq \lambda_k \leq ... \leq \lambda_n$

$$R(A, v) = \frac{\sum_{i=k}^{n} a_i^2 \lambda_i}{\sum_{i=k}^{n} a_i^2} \geq \lambda_k$$

and therefore

$$\max\{R(A, x) : x \in U\} \geq \lambda_k.$$

Since this is true for all subspaces U, we have that

$$\min_U\{\max_x\{R(A, x) : x \in U, x \neq 0\} : dim(U) = k\} \geq \lambda_k.$$

This is one inequality, we now find the other one. We now choose the k-dimensional space $V = \{u_1, ..., u_k\}$ where u_i is an orthonormal eigenvector of A where u_i corresponds to the λ_i eigenvalue of A. We use the the same arguments as in the previous equality. Take any vector $v \in V$ that intersects with U which is a subspace of dimension k, it must be a linear combination of basis

$$v = \sum_{i=1}^{k} a_i u_i, \quad a_i \in \mathbb{R}.$$

We have that

$$R(A, v) = \frac{(\sum_{i=1}^{k} a_i u_i)^\top A(\sum_{i=1}^{k} a_i u_i)}{(\sum_{i=1}^{k} a_i u_i)^\top (\sum_{i=1}^{k} a_i u_i)} = \frac{\sum_{i=1}^{k} a_i^2 \lambda_i}{\sum_{i=1}^{k} a_i^2} \leq \lambda_k,$$

so

$$\max_v\{R(A, v) : v \in V\} \leq \lambda_k$$

since λ_k is the largest eigenvalue in V. Therefore also,

$$\min_U\{\max_x\{R(A, x) : x \in U, x \neq 0\} : dim(U) = k\} \leq \lambda_k.$$

Taking these two inequalities we get that

$$\min_U\{\max_x\{R(A, x) : x \in U, x \neq 0\} : dim(U) = k\} = \lambda_k.$$

To prove the last statements

$$\max_x R(A, x) = \lambda_n, \quad \min_x R(A, x) = \lambda_1.$$

Take U with dimension one, the smallest value of the Rayleigh quotient will be equal to the smallest eigenvalue of A. Now also, take U with dimension n, the greatest value of the Rayleigh quotient will be equal to the greatest eigenvalue of A. □

2.3 Cayley-Hamilton Theorem

In this section we will establish the Cayley-Hamilton Theorem. But we need some other preliminaries first.

Definition 2.4. Let $A \in \mathbb{R}^{n\times n}$. The cofactor matrix C (or Minor) of A is a matrix where each entry C_{ij} of the cofactor matrix is a determinant of submatrix of A. This submatrix of A is formed by deleting the i-th row and the j-th column from A, and it is also multiplied by $(-1)^{i+j}$.

The transpose of the cofactor matrix is also known as the adjugate (or classical adjoint) of A, denoted $adj(A)$.

From *Cramer's Rule* we have that

$$A\ adj(A) = adj(A)\ A = det(A)I.$$

Theorem 2.2 (Cayley-Hamilton Theorem)**.** *Let $A \in \mathbb{R}^{n\times n}$ and $p(s) = \det(A - sI) = a_0 + a_1 s + \cdots + a_{n-1}s^{n-1} + a_n s^n$ be the characteristic polynomial of A. Then the matrix A satisfies its own characteristic polynomial, meaning that*

$$p(A) = a_0 I + a_1 A + \cdots + a_{n-1}A^{n-1} + a_n A^n = 0.$$

The following proof is based on material from [9].

Proof. Let $C(s)$ be the adjoint of $(A - sI)$. Since the entries of C are cofactors of $(A - sI)$ they are polynomials of at most degree $n-1$. Thus $C_{ij}(s) = c_0 + c_1 s + ... + c_{n-1}s^{n-1}$. Let $C(s) = B_0 + B_1 s + ... + B_{n-1}s^{n-1}$ where $B_i \in \mathbb{R}^{n\times n}$ are constant matrices.

From Cramer's rule we have that

$$(A - sI)\ adj(A - sI) = det(A)I \Leftrightarrow$$

$$\Leftrightarrow (A - sI)(B_0 + B_1 s + ... + B_{n-1}s^{n-1}) = (a_0 + a_1 s + \cdots + a_{n-1}s^{n-1} + a_n s^n)I.$$

We expand the left hand side and get

$$AB_0 + AB_1 s + ... + AB_{n-1}s^{n-1} - B_0 s - B_1 s^2 - ... - B_{n-1}s^n =$$

$$= (a_0 + a_1 s + \cdots + a_{n-1}s^{n-1} + a_n s^n)I.$$

Now, we get $n+1$ equations

$$-B_{n-1} = a_n I,$$

$$AB_{n-1} - B_{n-2} = a_{n-1}I,$$

$$AB_{n-2} - B_{n-3} = a_{n-2}I,$$

$$\ldots,$$

$$AB_1 - B_0 = a_1 I,$$

$$AB_0 = a_0 I.$$

Now if we multiply the first equation by A^n and the second by A^{n-1}, and we continue this for the rest of the equations. We get

$$-A^n B_{n-1} = a_n A^n,$$

$$A^n B_{n-1} - A^{n-1} B_{n-2} = a_{n-1} A^{n-1} \Leftrightarrow -a_n A^n - A^{n-1} B_{n-2} = a_{n-1} A^{n-1},$$

$$A^{n-1} B_{n-2} - A^{n-2} B_{n-3} = a_{n-2} A^{n-2} \Leftrightarrow$$

$$\Leftrightarrow -a_n A^n - a_{n-1} A^{n-1} - A^{n-2} B_{n-3} = a_{n-2} A^{n-2} \Leftrightarrow$$
$$\Leftrightarrow -a_n A^n - a_{n-1} A^{n-1} - a_{n-2} A^{n-2} = A^{n-2} B_{n-3} \Leftrightarrow$$
$$\Leftrightarrow -(\sum_{i=n-2}^{n} a_i A^i) = A^{n-2} B_{n-3},$$

$$\dots,$$

$$A^2 B_1 - AB_0 = a_1 A \Leftrightarrow -(\sum_{i=2}^{n} a_i A^i) - a_1 A = AB_0,$$

$$AB_0 = a_0 I \Leftrightarrow -(\sum_{i=1}^{n} a_i A^i) = a_0 I \Leftrightarrow 0 = a_0 I + a_1 A + a_2 A^2 + ... + a_n A^n.$$

So, the last equation shows the equality we wanted to derive. □

2.4 Matrix calculus

Matrix calculus is a specialized branch of calculus that deals with derivatives and integrals involving matrices and vectors. It extends the principles of traditional calculus to handle multidimensional objects. In control theory, matrix calculus is essential because it allows for differentiation and integration functions involving matrices of matrices and vectors that represent the dynamic behavior of systems.

2.4.1 Scalar-by-matrix derivation

Definition 2.5. If f is any scalar function and X is an $m \times n$ variable matrix, we define function-by-matrix derivative as

$$\frac{\partial f}{\partial X} = \begin{pmatrix} \partial f / \partial x_{11} & \dots & \partial f / \partial x_{1n} \\ \vdots & \ddots & \vdots \\ \partial f / \partial x_{m1} & \dots & \partial f / \partial x_{mn} \end{pmatrix}.$$

Definition 2.6. The trace of a matrix $A \in \mathbb{R}^{n \times n}$ is defined as

$$tr(A) = \sum_{i=1}^{n} A_{ii}.$$

We develop some notation for referencing the elements in matrices. For the ik-th element in the matrix $[AB]$ will be the i-th row of $A \in \mathbb{R}^{n\times m}$ multiplied by the k-th column of $B \in \mathbb{R}^{m\times p}$. So

$$[AB]_{ik} = \sum_{j=1}^{m} A_{ij} B_{jk}.$$

We can even extend this to 3 matrices, where $A \in \mathbb{R}^{n\times m}, B \in \mathbb{R}^{m\times p}, C \in \mathbb{R}^{p\times n}$ that are multiplied with each other

$$[ABC]_{il} = \sum_{j=1}^{m} A_{ij} [BC]_{jl} = \sum_{j=1}^{m} A_{ij} \sum_{k=1}^{p} B_{jk} C_{kl} = \sum_{j=1}^{m} \sum_{k=1}^{p} A_{ij} B_{jk} C_{kl}$$

where $C \in \mathbb{R}^{p\times}$

The two following proposition is based upon notes from [1].

Proposition 2.8. *Assume X is an $m \times p$ variable matrix and $A \in \mathbb{R}^{n\times m}, B \in \mathbb{R}^{p\times n}$. Then*

$$\frac{\partial tr(AXB)}{\partial X} = A^\top B^\top,$$

and if X is an $m \times p$ variable matrix and $A \in \mathbb{R}^{n\times p}, B \in \mathbb{R}^{m\times n}$

$$\frac{\partial tr(AX^\top B)}{\partial X} = BA.$$

Proof. For the first statement we have that

$$tr(AXB) = \sum_{i=1}^{n} [AXB]_{ii} = \sum_{i=1}^{n} \sum_{j=1}^{m} A_{ij} [XB]_{ji} = \sum_{i=1}^{n} \sum_{j=1}^{m} A_{ij} \sum_{k=1}^{p} X_{jk} B_{ki} =$$

$$= \sum_{i=1}^{n} \sum_{j=1}^{m} \sum_{k=1}^{p} A_{ij} X_{jk} B_{ki}.$$

Now if we take the derivative with respect to X_{jk}, all the terms without X_{jk} will disappear, so we get that

$$\frac{\partial tr(AXB)}{\partial X_{jk}} = \sum_{i=1}^{n} A_{ij} B_{ki} = [BA]_{kj}.$$

So, the jk-th element in the in derivative matrix would be kj-th element of $[BA]$, which is equivalent of the jk-th element of its transpose

$$[BA]^\top = A^\top B^\top.$$

Therefore

$$\frac{\partial tr(AXB)}{\partial X} = A^\top \ B^\top .$$

Now similarly for the second formula, and to remind $A \in \mathbb{R}^{n\times p}, B \in \mathbb{R}^{m\times n}$

$$tr(AX^\top B) = \sum_{i=1}^{n} [AX^\top B]_{ii} = \sum_{i=1}^{n}\sum_{j=1}^{p} A_{ij}[X^\top B]_{ji} = \sum_{i=1}^{n}\sum_{j=1}^{p} A_{ij} \sum_{k=1}^{m} X^\top_{jk} B_{ki} =$$

$$= \sum_{i=1}^{n}\sum_{j=1}^{p}\sum_{k=1}^{m} A_{ij} X_{kj} B_{ki}.$$

We take the derivative with respect to X_{kj}

$$\frac{\partial tr(AX^\top B)}{\partial X_{kj}} = \sum_{i=1}^{n} A_{ij} B_{ki} = [BA]_{kj}.$$

So the kj-th element of the derivative matrix is the kj-th element of $[BA]$. Therefore

$$\frac{\partial tr(AX^\top B)}{\partial X} = BA.$$

□

Proposition 2.9. *Let $A \in \mathbb{R}^{n\times m}$ and X be an $m \times n$ variable matrix. Then*

$$\frac{\partial tr(AX^\top)}{\partial X} = A.$$

Proof. We have that

$$tr(AX^\top) = \sum_{i=1}^{n} [AX^\top]_{ii} = \sum_{i=1}^{n}\sum_{j=1}^{m} A_{ij} X^\top_{ji} = \sum_{i=1}^{n}\sum_{j=1}^{m} A_{ij} X_{ij}.$$

Now if we derivate the function by X_{ij}

$$\frac{\partial (A_{ij} X^\top_{ji})}{\partial X_{ij}} = \frac{\partial (A_{ij} X^\top_{ij})}{\partial X_{ij}} = A_{ij}.$$

□

Proposition 2.10. *Let $A \in \mathbb{R}^{m\times n}$ and X be an $m \times n$ variable matrix. Then*

$$\frac{\partial tr(A^\top X)}{\partial X} = A.$$

Proof. We do a similar proof as the previous proposition

$$tr(A^\top X) = \sum_{i=1}^{n} [A^\top X]_{ii} = \sum_{i=1}^{n} \sum_{j=1}^{m} A^\top_{ij} X_{ji} = \sum_{i=1}^{n} \sum_{j=1}^{m} A_{ji} X_{ji}.$$

Now if we derivate the function by X_{ji}

$$\frac{\partial (A_{ji} X_{ji})}{\partial X_{ji}} = A.$$

□

Proposition 2.11. *Let and X be an $n \times n$ variable matrix. Then*

$$\frac{\partial tr(X)}{\partial X} = I.$$

Proof. We have

$$tr(X) = \sum_{i=1}^{n} X_{ii}.$$

So when

$$j = i: \quad \frac{\partial (\sum_{i=1}^{n} X_{ii})}{\partial X_{ii}} = 1, \quad \text{and} \quad i \neq j: \quad \frac{\partial (\sum_{i=1}^{n} X_{ii})}{\partial X_{ij}} = 0$$

therefore

$$\frac{\partial tr(X)}{\partial X} = I.$$

□

Proposition 2.12. *Let $A \in \mathbb{R}^{n \times m}$ and X be an $n \times m$ variable matrix. Then*

$$\frac{\partial tr(AX^\top XA^\top)}{\partial X} = 2XA^\top A.$$

Proof. According to [1] in (18) if we have multiple occurrences of X in the trace function, we can simply evaluate each appearance of X assuming that everything else is constant (including other appearances of X) and then summing those evaluations. We have that

$$\frac{\partial tr(AX^\top XA^\top)}{\partial X} = \frac{\partial tr(AX^\top D)}{\partial X} + \frac{\partial tr(EXA^\top)}{\partial X}$$

where $D = XA^\top$ and $E = XA^\top$. Now, by proposition (2.8) we have that

$$\frac{\partial tr(AX^\top D)}{\partial X} + \frac{\partial tr(AX^\top E)}{\partial X} = DA + E^\top A = (XA^\top)A + (XA^\top)A = 2XA^\top A.$$

□

2.4.2 Scalar-by-vector derivation

Definition 2.7. If y is a real variable (that might depend on several variables) and x is a variable vector with n components, we define scalar-by-vector derivatives as

$$\frac{\partial y}{\partial x} = \begin{pmatrix} \partial y/\partial x_1 & \dots & \partial y/\partial x_n \end{pmatrix}.$$

Proposition 2.13. *Let $A \in \mathbb{R}^{n\times n}$ and x be an n variable vector. Then the derivative of the quadratic form*

$$x^\top Ax$$

is

$$\frac{\partial x^\top Ax}{\partial x} = x^\top (A + A^\top).$$

Proof. By definition

$$x^\top Ax = \sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij}x_ix_j.$$

Now, if we derivate with respect to the $k-th$ element of x we have that

$$\frac{\partial x^\top Ax}{\partial x_k} = \sum_{j=1}^{n} a_{kj}x_j + \sum_{i=1}^{n} a_{ik}x_i$$

for all $k = 0, 1, ..., n$. We denote A_k as the k-th column in A. In the first term $\sum_{j=1}^{n} a_{kj}x_j$ this would be equivalent of $x^\top A_k = \sum_{j=1}^{n} a_{kj}x_j$. For the second term $\sum_{i=1}^{n} a_{ik}x_i = x^\top A_k^\top$.

So,

$$\frac{\partial x^\top Ax}{\partial x_k} = \frac{\partial(\sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij}x_ix_j)}{\partial x_k} = \sum_{j=1}^{n} a_{kj}x_j + \sum_{i=1}^{n} a_{ik}x_i =$$

$$= x^\top A_k + x^\top (A^\top)_k = x^\top (A_k + (A^\top)_k).$$

therefore

$$\frac{\partial x^\top Ax}{\partial x} = x^\top (A + A^\top).$$

□

Proposition 2.14. *Let $A \in \mathbb{R}^{n\times n}$ be a symmetric matrix and x be an n variable vector. Then the derivative of the quadratic form*

$$x^\top Ax$$

is

$$\frac{\partial x^\top Ax}{\partial x} = 2x^\top A.$$

Proof. his is simply an application of the previous proposition. We know that for any matrix A

$$\frac{\partial x^\top Ax}{\partial x} = x^\top (A + A^\top).$$

Now since $A = A^\top$ and $A + A^\top = 2A$, we have that

$$x^\top (A + A^\top) = 2x^\top A.$$

□

Proposition 2.15. *Let $A \in \mathbb{R}^{n\times n}, \quad y \in \mathbb{R}^n$ and x be a n variable vector. Then*

$$\frac{\partial y^\top Ax}{\partial x} = y^\top A.$$

Proof. Let $v^\top = y^\top A$ and denote the the i-th element of v as

$$v_i = \sum_{j=1}^{n} y_j A_{ij}.$$

We have that

$$\frac{\partial v_i x_i}{\partial x_i} = \frac{\partial \sum_{j=1}^{n} y_j A_{ij} x_i}{\partial x_i} = \sum_{j=1}^{n} y_j A_{ij}.$$

□

Proposition 2.16. *Let $A \in \mathbb{R}^{n\times n}, \quad y \in \mathbb{R}^n$ and x be a n variable vector. Then*

$$\frac{\partial x^\top Ay}{\partial x} = y^\top A^\top.$$

Proof. Let $v = Ay$ and denote the the i-th element of v as

$$v_i = \sum_{j=1}^{n} A_{ji} y_j.$$

We have that

$$\frac{\partial x_i v_i}{\partial x_i} = \frac{\partial \sum_{j=1}^{n} x_i A_{ji} y_j}{\partial x_i} = \sum_{j=1}^{n} y_j A_{ji}.$$

□

2.5 Vector- and Matrix Norms

Norms enable us to define distances between vectors or matrices. By calculating the norm of the difference between two vectors or matrices, we obtain a distance metric that quantifies how "far apart" they are. This enabled us to compare vectors or matrices for analyzing their behaviors and properties under different circumstances.

Definition 2.8. Let $x \in \mathbb{R}^n$ and $M \in \mathbb{R}^{n\times n}$, the Euclidean vector norm and the Frobenius matrix norm is respectively defined as

$$||x||_2 = \sqrt{\sum_{i=1}^{n} x_i^2}, \quad ||M||_F = \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{n} M_{ij}^2}.$$

Proposition 2.17. *Let $M \in \mathbb{R}^{n\times n}$. Then*

$$||M||_F = \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{n} M_{ij}^2} = \sqrt{tr(MM^\top)}.$$

Proof. First we have that the entries of $MM^\top$ is

$$[MM^\top]_{ik} = \sum_{j=1}^{n} M_{ij}M_{jk}^\top = \sum_{j=1}^{n} M_{ij}M_{kj}.$$

The trace of $MM^\top$ is

$$tr([MM^\top]) = \sum_{i=1}^{n}[MM^\top]_{ii} = \sum_{i=1}^{n}\sum_{j=1}^{n} M_{ij}M_{ji}^\top = \sum_{i=1}^{n}\sum_{j=1}^{n} M_{ij}M_{ij} = \sum_{i=1}^{n}\sum_{j=1}^{n} M_{ij}^2.$$

Now of we simply take the root of $tr([MM^\top])$ we have the result. □

Another important matrix norm is the spectral norm. Which is an induced norm from the Euclidean vector norm. We first show that the Euclidean vector norm preserves distance. Then we show how the Spectral norm is induced from the Euclidean vector norm.

Proposition 2.18. *Let $M \in \mathbb{R}^{n\times n}$ be an orthonormal matrix. Then*

$$||Mx||_2 = ||x||_2,$$

in other words the Euclidean vector norm is distance preserving.

Proof. Since M is an orthonormal matrix $M^\top M = MM^\top = I$, we have that

$$||Mx||_2 = \sqrt{(Mx)^\top Mx} = \sqrt{x^\top M^\top Mx} = \sqrt{x^\top x} = ||x||_2.$$

□

Now we show how to induce the Spectral norm. Let $A \in \mathbb{R}^{n\times n}$, recall the Singular Value Decomposition (SVD) of A, i.e. $A = U\Sigma V^\top$ where $U, V \in \mathbb{R}^{n\times n}$ are orthonormal matrices and $\Sigma \in \mathbb{R}^{n\times n}$ is a diagonal matrix with the singular values of A along its diagonal in decreasing order starting from the first element.

Proposition 2.19. *Let $M \in \mathbb{R}^{n\times n}$ and the singular values of M be $\sigma_1 \leq ... \leq \sigma_k$ where k is the dimension of M. Then*

$$\max_x \frac{||Mx||_2}{||x||_2} = \max_{||x||_2=1} ||Mx||_2 = \sigma_k.$$

Proof. We have that

$$\max_{||x||_2=1} ||Mx||_2 = \max_{||x||_2=1} ||U\Sigma V^\top x||_2 = \max_{||x||_2=1} ||\Sigma V^\top x||_2.$$

Now since $||V^\top x||_2 = ||x||_2 = 1$ and the expression $\Sigma V^\top x = \Sigma y$ is maximized when Σ is maximized. This happens when the first element of y is equal to one and the other elements is equal to zero in other words $y = (1\ 0\ ...\ 0)^\top$. This because $||V^\top x||_2 = 1$. We rewrite

$$\max_{||x||_2=1} ||\Sigma V^\top x||_2 = \max_{||y||_2=1} ||\Sigma y||_2 = \sigma_k.$$

The vector x that maximize

$$\max_{||x||_2=1} ||\Sigma V^\top x||_2$$

is the first row of $V^\top$. Denote v_i as the i-th row of $V^\top$. Now if $x = v_1^\top$, then $v_i \cdot x = 0$ for all $i \neq 1$ and $v_1\ v_1^\top = 1$ because the rows and columns of $V^\top$ are orthonormal. So the vector $x = v_1^\top$ maximize the the original expression

$$\max_{||x||_2=1} ||Mx||_2.$$

□

Definition 2.9. Let $M \in \mathbb{R}^{n\times n}$ and the singular values of M is $\sigma_1 \leq ... \leq \sigma_k$, where k is the dimension of M. The *Spectral norm* of M is defined as

$$||M||_2 := \max_x \frac{||Mx||_2}{||x||_2} = \sigma_k.$$

Proposition 2.20. *Let $M \in \mathbb{R}^{n \times n}$ be an invertible matrix and the singular values of M is $\sigma_1 \leq ... \leq \sigma_n$. Then*

$$||M^{-1}||_2 = \frac{1}{\sigma_1}.$$

Proof. We use a similar argument done in the previous proposition. We have that

$$M^{-1} = (U\Sigma V^\top)^{-1} = V\Sigma^{-1}U^\top.$$

Now we have that

$$\max_{||x||_2=1} ||V\Sigma^{-1}U^\top x||_2 = \max_{||x||_2=1} ||\Sigma^{-1}U^\top x||_2.$$

Now since $||U^\top x||_2 = ||x||_2 = 1$ and the expression $\Sigma^{-1}U^\top x = \Sigma^{-1}y$ is maximized when Σ^{-1} is maximized. This happens when the last element of y is equal to one and the other elements is equal to zero in other words $y = (0\ 0\ ...\ 1)^\top$. This because $||U^\top x||_2 = 1$.

We rewrite

$$\max_{||x||_2=1} ||\Sigma^{-1}U^\top x||_2 = \max_{||y||_2=1} ||\Sigma^{-1}y||_2 = \frac{1}{\sigma_1}.$$

The vector x that maximize

$$\max_{||x||_2=1} ||\Sigma^{-1}U^\top x||_2$$

is the first row of $U^\top$. Denote u_i as the i-th row of $U^\top$. Now if $x = u_n^\top$, then $v_i \cdot x = 0$ for all $i \neq 1$ and $u_n\ u_n^\top = 1$ because the rows and columns of $U^\top$ are orthonormal. So the vector $x = u_n^\top$ maximize the the original expression

$$\max_{||x||_2=1} ||M^{-1}x||_2.$$

□

Proposition 2.21. *Let $A, B \in \mathbb{R}^{n \times n}$. Then*

$$||AB||_2 \leq ||A||_2||B||_2.$$

Proof. From the definition of the Spectral norm we have that

$$||A||_2 = \max_x \frac{||Ax||_2}{||x||_2},$$

so it must be that

$$||A||_2 \geq \frac{||Ax||_2}{||x||_2} \Leftrightarrow ||A||_2||x||_2 \geq ||Ax||_2.$$

We have that

$$||ABx||_2 \leq ||A||_2||Bx||_2 \leq ||A||_2||B||_2||x||_2.$$

Now if we set the condition $||x||_2 = 1$ and

$$\max_{||x||_2=1} ||ABx||_2 \leq ||A||_2||B||_2||x||_2 \Leftrightarrow ||AB||_2 \leq ||A||_2||B||_2.$$

□

3 Control Theory and Dynamical systems

The desire to comprehend and control the behavior of complex systems drives the study of control theory. Control theory is fundamental to contemporary engineering and technology, including applications in aircraft, robotics, manufacturing, and energy systems, by giving a systematic framework for creating, evaluating, and putting into practice control systems. Using control theory, researchers can describe and comprehend the behavior of complicated systems, forecast how they will react to various inputs, and create controllers that will provide the desired results.

Control theory is a broad and varied topic of study that has numerous chances for research and invention. In conclusion, learning about control theory may provide one a thorough grasp of the ideas that guide current technology and open doors for creativity and research in a variety of sectors.

Dynamical systems are systems that evolve over time according to certain mathematical models or equations. A common description of a dynamical system is that we have some differential equation

$$\dot{x}(t) = A(t)x(t) + B(t)u(t)$$

where $A_i \in \mathbb{R}^{n\times n}, x \in \mathbb{R}^n, B_i \in \mathbb{R}^{n\times m}$ and with some control input $u_i \in \mathbb{R}^m$. This is usually known as a continuous-time.

Similarly, we can also define a discrete-time dynamical system

$$x(k+1) = A(k)x(k) + B(k)u(k).$$

Since many problems in this paper are discrete, linear and time-invariant this, we will mostly be studying Linear time-invariant (LTI) discrete dynamical systems, meaning that the matrices A, B are constant. So this formulation $x(k+1) = Ax(k) + Bu(k)$ will be the most common one we will study.

The following material in this section is heavily based upon several sources, communication and explanations with supervisor [5], Mathematical Control Theory by Eduardo D Sontag [7], lecture notes and other material from the course [9].

3.1 Reachability (Discrete time)

Let put the abstract notion of reachability in the the discrete-time linear system

$$x(k+1) = A(k)x(k) + B(k)u(k), \; k \geq k_0$$

where $A(k) \in \mathbb{R}^{n\times n}$, $B(k) \in \mathbb{R}^{n\times m}$, $x(k) \in \mathbb{R}^n$ and $x(k) \in \mathbb{R}^n$, and k, k_0 are integers. Clearly

$$x(k) = \Phi(k, k_0)x(k_0) + \sum_{i=k_0}^{k-1} \Phi(k, i+1)B(i)u(i)$$

where the state transition matrix $\Phi(k, k_0) = A(k-1)A(k-2)\cdots A(k_0)$, for $k > k_0$ and $\Phi(k_0, k_0) = I$. Let now the initial state at time k_0 be $x_0 = x(k_0)$. For the state at some time $k_1 > k_0$ to assume the value x_1, an input $u(\cdot)$ must exist such that

$$x_1 = \Phi(k_1, k_0)x_0 + \sum_{i=k_0}^{k_1-1} \Phi(k, i+1)B(i)u(i).$$

For the purpose of this text and the sake of notational simplicity, we restrict ourselves to the time-invariant linear system, that is, $A(k), B(k)$ are constant matrices for all $k \geq k_0$. The derivation is similar for the time varying system. Then $\Phi(k_1, k_0) = A^{k-k_0}$ depends on $k - k_0$ and we can simply take $k_0 = 0$, and $k_1 = K$. Thus

$$x_1 = A^K x_0 + \sum_{i=0}^{K-1} A^{K-(i+1)}Bu(i), \; K > 0$$

which is

$$x_1 = A^K x_0 + \begin{pmatrix} B & AB & \cdots & A^{K-1}B \end{pmatrix} \begin{pmatrix} u(K-1) \\ u(K-2) \\ \vdots \\ u(0) \end{pmatrix} = A^K x_0 + R_K(A, B)U_K$$

where $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, and $R_K(A, B) = \begin{pmatrix} B & AB & \cdots & A^{K-1}B \end{pmatrix}$, and $U_K = \begin{pmatrix} u(K-1)^\top & u(K-2)^\top \cdots & u(0)^\top \end{pmatrix}^\top$. In the sequel we will use $R(A, B)$ if $K = n$.

The question of the reachability is: Given x_0 at $k = 0$ and x_1 at $k = K > 0$ is there a input sequence $u(\cdots)$ (in other words the vector U_K such that $x_1 = A^K x_0 + R(A, B)U_K$? The answer is that such U_K exists if $x_1 - A^K x_0$ is in the range of $R(A, B)$. So we proved the following theorem.

Theorem 3.1. *There exists input $u(k)$ that transfers the state of the $x(k+1) = Ax(k) + Bu(k)$ from x_0 to x_1 in some finite time if and only if*

$$x_1 - A^K x_0 \in \mathcal{R}_K(A, B),$$

i.e. x_1 lies in the range of $R_K(A, B)$. Such input $u(k)$, $k = 0, 1, ..., K - 1$ is determined by solving the equation

$$R_K(A, B)U_K = x_1 - A^K x_0.$$

Now we rephrase the notion of reachability for the time-invariant discrete-time linear system. A state x_1 is *reachable* if there exists a $u(k)$, $0 \leq k \leq K$ that drives the state $x(k)$ from x_0 at $k = 0$ to x_1 in some finite time K. Denote R_r the set of all reachable states of the system $x(k + 1) = Ax(k) + Bu(k)$, which is a vector space so we call R_r the reachable subspace of the system $x(k + 1) = Ax(k) + Bu(k)$, We say $x(k + 1) = Ax(k) + Bu(k)$, is completely reachable if every state is reachable, i.e. $R_r = \mathbb{R}^n$. Since this only depends on the pair (A, B) we simply say that A, B is reachable. Now we are in position to prove the following theorem.

Next we show that the transfer take at most n steps, the dimension of the system.

Theorem 3.2. There exists input $u(k)$ that transfers the state of the $x(k+1) = Ax(k) + Bu(k)$ from $x_0 = 0$ to x_1 in finite time if and only if

$$x_1 \in \mathcal{R}_K(A, B),$$

i.e. x_1 lies in the range of $R_K(A, B)$. Moreover an appropriate input $u(k)$, $k = 0, 1, ..., n - 1$ that accomplishes this transfer in n steps is determined by

$$U_n = \begin{pmatrix} u(n-1)^\top & u(n-2)^\top \cdots & u(0)^\top \end{pmatrix}^\top$$

which is a solution to the equation

$$R(A, B)U_n = x_1.$$

In this case, x_1 is reachable and $R_r = \mathcal{R}(A, B)$.

Proof. Since we have already proved that such a transfer exists if and only if $x_1 - A^K x_0 = x_1 \in \mathcal{R}_K(A, B)$, or $x_1 = R_K(A, B)U_K$ has a solution U_K, it remains to show that it takes n steps to accomplish this transfer.

For x_1 to be reachable we must have $x_1 \in \mathcal{R}_K(A, B)$ for some finite K. Note that the range $\mathcal{R}_K(A, B)$ cannot increase beyond the range of $\mathcal{R}_n(A, B) = \mathcal{R}(A, B)$, that is, $\mathcal{R}_K(A, B) = \mathcal{R}(A, B)$ for $K \geq n$. This is a consequence of the Cayley-Hamilton Theorem (Theorem 2.2), because any vector x in $\mathcal{R}_K(A, B)$ $K \geq n$, can be expressed as a linear combination of $B, AB, ..., A^{n-1}B$. Therefore, $x \in \mathcal{R}(A, B)$. It is possible to have $x_1 \in \mathcal{R}_K(A, B)$ with $K < n$ for a particular x_1. However, in this case $x_1 \in \mathcal{R}(A, B)$ since $\mathcal{R}_K(A, B)$ is a subset of $\mathcal{R}(A, B)$. Hence, x_1 is reachable if and only if it is in the range of $\mathcal{R}(A, B)$. Clearly any U_n that accomplishes this transfer satisfies the equation $R(A, B)U_n = x_1$. □

Corollary: *The system $x(k+1) = Ax(k) + Bu(k)$ is completely reachable (or the pair (A, B) is reachable) if and only is*

$$\operatorname{rank} R(A, B) = n.$$

Proof. It is an immediate consequence of the preceding theorem by noting that $\mathcal{R}(A, B) = R_r = \mathbb{R}^n$ if and only if the rank of $R(A, b)$ is equal to n. □

Remark. This holds for is any initial state x_0. The proof is similar.

The following notion will be used later.

Definition 3.1. The reachability gramian of the system $x(k+1) = Ax(k) + Bu(k)$ is defined by

$$W_r(0, K) = \sum_{i=0}^{K-1} A^{K-(i+1)} BB^\top (A^\top)^{K-(i+1)}.$$

Note that

$$\sum_{i=0}^{K-1} A^{K-(i+1)} BB^\top (A^\top)^{K-(i+1)} = \sum_{i=0}^{K-1} A^i BB^\top (A^\top)^i = R_K(A, B) R_K(A, B)^\top$$

we have

Proposition 3.1. The following holds for the reachability gramian satisfies

$$W_r(0, K) = R_K(A, B) R_K(A, B)^\top.$$

Proposition 3.2. $\mathcal{R}(A, B) = \mathcal{R}(W_r(0, K))$ for all $K \geq 0$.

Proof. Take $x_1 \in \mathcal{R}(W_r(0, K))$. Then there is an $\eta_1 \in \mathbb{R}^n$ such that $W_r(0, K)\eta_1 = x_1$. But $x_1 = \sum_{i=0}^{K-1} A^{K-(i+1)} Bu(i)$. Choose

$u(i) = B^\top (A^\top)^{K-(i+1)} \eta_1$ we have

$$x_1 = W_r(0, K)\eta_1 = \left(\sum_{i=0}^{K-1} A^{K-(i+1)} BB^\top (A^\top)^{K-(i+1)} \right) \eta_1$$

which means $x_1 \in \mathcal{R}_K(A, B)$, proving $\mathcal{R}(W_r(0, K)) \subseteq \mathcal{R}(A, B)$.

On the other hand, if $x_1 \in \mathcal{R}_K(A, B)$, i.e., there exists $\eta \in \mathbb{R}^{m \times n}$ such that $R_K(A, B)\eta = x_1$. Assuming $x_1 \notin \mathcal{R}(W_r(0, K))$ for some $K > 0$. Then $\mathcal{N}(W_r(0, K))$ is nontrivial. So there is $x_2 \neq 0$ such that $x_2^\top x_1 \neq 0$, that is x_1 and x_2 are not orthogonal. To see this note first that $W_r(0, K)$ is symmetric which leads to $\mathcal{R}(W_r(0, K)) = (\mathcal{N}(W_r(0, K))^\perp$. Since $W_r(0, K)x_2 = 0$ we have x_1 such that $x_2^\top x_1 = 0$ would be in the range of $W_r(0, K)$, which is not true, and so $x_2^\top x_1 \neq 0$.

Next consider

$$x_2^\top W_r(0,K)x_2 = 0 = \sum_{i=0}^{K-1} \left(x_2^\top A^{K-(i+1)}B\right)\left(B^\top (A^\top)^{K-(i+1)}\right) = \sum_{i=0}^{K-1} \|x_2^\top A^{K-(i+1)}B\|_2^2$$

implying $x_2^\top A^{K-(i+1)}B = 0$ for all $0 \leq i \leq K$.

This in turn shows that

$$x_2^\top A^i B = 0, i \geq 0$$

Therefore $x_2^\top x_1 = x_2^\top R_K(A,B)\eta = 0$ which is a contradiction since $x_2^\top x_1 \neq 0$. Therefore x_1 lies in the range of $W_r(0,K)$, completing the proof. □

3.2 Optimal control

When applying some control it is reasonable to assume that it is not "free" to use it. It has some "cost" of using the control, whether it is fuel in some motor, financial or political cost of applying some economical policy. Finding the control inputs that produce the desired results while reducing some measure of cost or maximizing some measure of performance is the aim of Optimal control.

3.2.1 Performance indices

Depending on our goals or objectives how we measure performance might differ. Performance indices quantify how good our control function performs. We present some common performance indices.

Assume our dynamical system is subject to some to some initial condition

$$x(t_0) = x_0.$$

In many problems we want to know how good our system is performing according so measurement. A *performance index* **J** is scalar valued function which provides a measurement of performance on our system.

In **Minimum time problems** we want to transfer from our initial state $x(t_0)$ to $x(t_1) = x_{t_f}$ where x_{t_f} is our final state which is specified, in an minimum time. A suitable performance index for this problem would be

$$J = \int_{t_0}^{t_1} dt = t_1 - t_0,$$

a discrete version would be

$$J = \sum_{i=t_0}^{t_1} 1 = t_1 - t_0.$$

In **Minimum effort problems** our final state is specified and we want to reach this state with as little control as possible. Here, a suitable performance index could either be a linear performance index

$$J = \int_{t_0}^{t_1} \sum_{j=1}^{m} \beta_j |u_j| dt,$$

or quadratic performance index

$$J = \int_{t_0}^{t_1} u^\top R u dt$$

where is R is real positive definite matrix and r_{ij}, β_i are weighting factors. The discrete version would simply be

$$J = \sum_{i=t_0}^{t_1} \sum_{j=1}^{m} \beta_j |u_j|,$$

and the quadratic performance index

$$J = \sum_{i=t_0}^{t_1} u^\top R u.$$

3.3 Dynamic programming (DP)

The optimality principle stated by Bellman is as follows:

An optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision.

Briefly we can state the Bellman's principle of optimality as follows. *From any point on an optimal trajectory, the remaining trajectory is optimal for the corresponding problem initiated at that point.* We illustrate this principle by an example.

The shortest path problem: Consider the "stagecoach problem" (drawn in a directed graph, see *Figure 4*) in which a traveler wishes to minimize the cost of a journey from an initial town (node) to a terminal town (node) through several possible paths. To each path is associated a cost shown on each arc in the path. An example is the uppermost path $0 \to 1 \to 1 \to 1 \to 1 \to 0$ has the cost $1 + 5 + 4 + 1 + 2 = 13$. The problem is to find the path with minimal cost. A naive (but natural) way to solve this problem is to compute the cost of each path and then compare them. If N is the number of stages (here $N = 5$ then we can show that there are $\mathcal{O}((1+\sqrt{2})^N)$ paths. Since we need to add numbers to compute the cost of a single path, we need $\mathcal{O}((1+\sqrt{2})^N N)$ additions in total and then compare $\mathcal{O}((1+\sqrt{2})^N)$ numbers in order to find the shortest path. This is a very large number when N is large. Dynamic programming provides a systematic and less expansive way to solve this problem. Let $k \in \{0, 1, 2, 3, 4, 5\}$ denote the stage and for each stage the state $x_k := x(k) \in \{1, 0, -1\}$ tells us

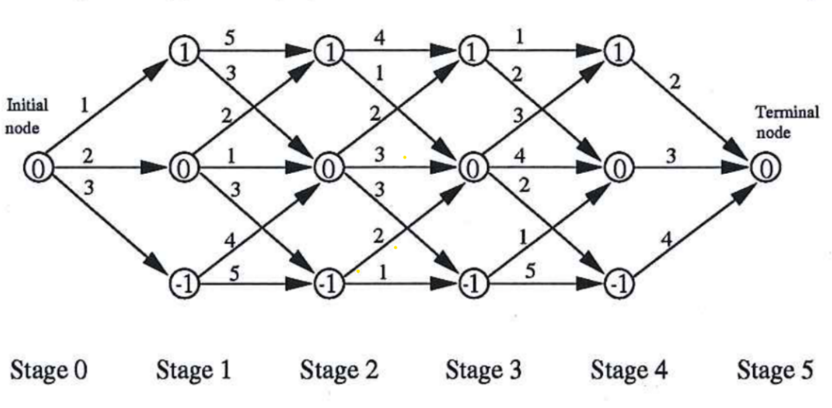

Figure 4: Road system for stagecoach problem

whether we are in the upper, middle or lower node respectively. In this way we can represent the nodes of the graph with their ”coordinates” (k, x_k). Let the shortest path (minimum cost path) from node (k, x) to the terminal node be $J(k, x)$. Then $J(0, 0)$ is the shortest path from stage 0 satisfying (obviously) the following relation

$$J(0,0) = \min\left(1 + J(1,1), 2 + J(1,0), 3 + J(1,-1)\right).$$

Continue in the same manner, for example

$$J(1,1) = \min\left(5 + J(2,1), 3 + J(2,0)\right).$$

The basic principle behind these formulas is the Bellman optimality principle: *The shortest path has the property that for any initial part of the path from the initial node to some node* $(k, x) \in \{1, ..., 5\} \times \{1, 0, -1\}$ *the remaining path must be the shortest from the node* (k, x) *to the terminal.*

Notice that the cost in the terminal of the shortest path is known in advance. In our example $J(5, 0) = 0$. This means we can optimize backwards from stage 5 to stage 0 and in this recursive way we can compute the *shortest path-to-go* function $J(k, x)$. Since there is only one way of going from the nodes are stage 4 to the terminal node we get

$$J(4,1) = 2,\ J(2,0) = 3,\ J(4,-1) = 4.$$

In the nest step we obtain

$$\begin{aligned}
J(3,1) &= \min\left(1+J(4,1), 2+J(4,0)\right) = \min(3,5) = 3\\
J(3,0) &= \min\left(3+J(4,1), 4+J(4,0), 2+J(4,-1)\right) = \min(5,7,6) = 5\\
J(3,-1) &= \min\left(1+J(4,0), 5+J(4,-1)\right) = \min(4,9) = 4.
\end{aligned}$$

Continue this way we can find the minimum cost $J(0,0) = 8$. The path is shown in *Figure 5* with the thicker arrows.

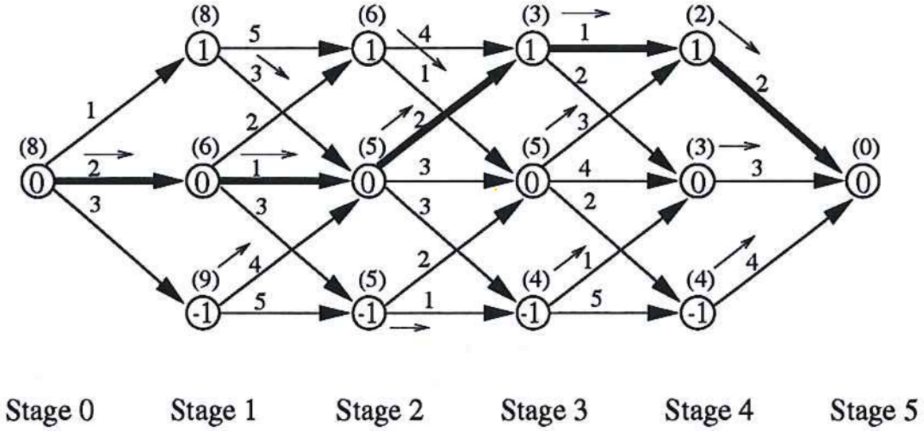

Figure 5: Optimal solution of stagecoach problem

We comment on the complexity of the dynamic programming approach now. No addition and comparison are needed while computing $J(5,x)$ and $J(4,x)$. To compute $J(3,x)$ for $x = 1, 0, -1$ we need 7 additions and 4 comparisons. Thus for arbitrary number of stages N we need 3+7(N-2) additions and 2+4(N-2) comparisons. For large N this is much less expansive than computing the cost of all possible paths and then comparing them although it is on the same order of complexity.

3.3.1 DP in discrete time

To avoid complication of technicality and get good intuition we first derive the optimality equation for discrete-time problem. The above shortest path problem is a special case of multistage decision problem. The general form is an optimal control problem:

$$(P_d) \quad \begin{cases} \text{minimize} \quad \phi(x_N) + \displaystyle\sum_{k=0}^{N-1} f_0(k, x_k, u_k) \\ \text{subject to} \quad x_{k+1} = f(k, x_k, u_k), x_0 \text{ is given }, x_k \in X_k, u_k \in U(k,x), \end{cases}$$

where $k \in \{0, 1, ..., N\}$ (we call it discrete-time set), X_k is the state space, a discrete set in the shortest path example but it is very often $X_k = \mathbb{R}^n$ for all $k = 0, 1..., N$, and $U(k, x_k)$ is the constraint set in $\mathbb{R}^m$. Note that the cost function is additive and has one term corresponding to each stage. The terminal cost $\phi(x_N)$ penalizes deviation from a desired terminal state and the running cost adds a term $f_0(k, x_k, u_k)$ to the total cost at each stage.

The preceding optimization problem can be generalized to $x_0 \in S_0$ and $x_N \in S_N$ where $S_0 \subset X_0$ and $S_N \subset X_N$ are the subsets of the state space. We can also let N be free (that is a variable).

Now we cast the shortest path example in this formulation.

- $x_0 = 0$, $X_k = \{1, 0, -1\}$ at $k = 1, ..., N-1$ and $X_N = \{0\}$.
- The control variable takes three values (in most cases) $1, 0, -1$ where $u_k = 1$ means going up, $u_k = 0$ going forward and $u_k = -1$ going down. The control constraint set is

$$U(k, x) = \begin{cases} \{0, .1\}, & x = 1 \\ \{1, 0, -1\}, & x = 0 \qquad k = 0, 1, ..., N-2 \\ \{1, 0\}, & x = -1 \end{cases}$$

$$U(N-1, 1) = \{-1\}, U(N-1, 0) = \{0\},\ U(N-1, -1) = \{1\}.$$

- the state dynamics is given by $x_{k+1} = x_k + u_k$, i.e. $f(k, x, u) = x + u$.
- the terminal cost $\phi(x) = 0$ and the stage-wise additive costs $f_0(k, x, u) = c^k_{i,j}$, where $c^k_{i,j}$ is the cost on the arrow from node (k, i) at stage k to node $(k+1, j)$ at stage $k+1$. For example, $c^0_{0,1} = 1$, $c^1_{1,1} = 5$, $c^1_{1,0} = 3$, etc..

3.3.2 The dynamic programming equation

Define the optimal *cost-to-go* function as

$$J^*(n, x) = \min \left\{ \phi(x_N) + \sum_{k=n}^{N-1} f_0(k, x_k, u_k) : \right.$$

$$\left. x_{k+1} = f(k, x_k, u_k), x_n = x, x_k \in X_k, u_k \in U(k, x_k) \right\}$$

for $n = 0, ..., N-1$ and $J^*(N, x) = \phi(x)$. In particular, the optimal solution of (P_d) is $J^*(0, x_0)$.

Theorem 3.3 (The principle of optimality)**.** *Suppose there is a finite solution to the backwards dynamic programming recursion*

$$J(N,x) = \begin{cases} \phi(x), & x \in X_N \\ \infty, & x \notin X_N \end{cases}$$
$$J(n,x) = \min\left\{f_0(n,x,u) + J(n+1, f(n,x,u))\right\}, \quad n = N-1, ..., 0$$

where the optimization over $U(n,x)$ is restricted to those control variables for which $f(n,x,u) \in X_{n+1}$. Then, there exists an optimal solution to (P_d) and

- *$J^*(n,x) = J(n,x)$ for all $n = 0, ..., N$, $x \in X_n$, and*
- *the optimal feedback control in each stage is*

$$u_n^* = \mu(n,x) = \arg\min_{u \in U(n,x)} \left\{f_0(n,x,u)0J(n+1, f(n,x,u))\right\}.$$

Proof. In fact, it is an immediate consequence of the principle of optimality. But we give a proof based on induction. First, we have $J^*(N,x) = J(N,x) = \phi(x)$. Assume now that for some $n \in \{1, ..., N-1\}$ we have $J^*(n+1,x) = J(n+1,x)$ for all $x \in X_{n+1}$. Then

$$\begin{aligned} J^*(n,x_n) &= \min_{u_k \in U(k,x_k), k=n,...,N-1} \left\{\phi(x_N) + \sum_{k=n}^{N-1} f_0(k,x_k,u_k)\right\} \quad \text{(by definition)} \\ &= \min_{u_n \in U(n,x_n)} \left\{f_0(n,x_n,u_n) + \min_{u_k \in U(k,x_k), k=n+1,...,N-1} \left\{\phi(x_N) + \sum_{k=n+1}^{N-1} f_0(k,x_k,u_k)\right\}\right\} \\ &\qquad \text{(by the additivity of the cost function)} \\ &= \min_{u_n \in U(n,x_n)} \left\{f_0(n,x_n,u_n) + J^*(n+1, f(n,x_n,u_n))\right\} \ (x_{n+1} = f(n,x_n,u_n)) \end{aligned}$$

Now by the induction assumption we obtain

$$J^*(n,x_n) = \min_{u_n \in U(n,x_n)} \left\{f_0(n,x_n,u_n) + J(n+1, f(n,x_n,u_n))\right\}.$$

completing the proof. □

The optimal cost-to-go function, $J^*(n,x)$ is commonly called *the value function.* Sometimes we use the notation $V(n,x)$. In this notation the dynamic programming equation is

$$(DP) \quad \begin{cases} V(N,x) = \begin{cases} \phi(x), & x \in X_N \\ \infty, & x \notin X_N \end{cases} \\ V(n,x) = \min\left\{f_0(n,x,u) + V(n+1, f(n,x,u))\right\}, \quad n = N-1, ..., 0. \end{cases}$$

Keep the following in mind.

(i) The optimal u_k is a function of x_k and k, i.e., $u_k = \mu(k, x_k)$

(ii) The (DP) equation yields the optimal control u_k in closed loop form. It is optimal whatever the past control policy may have been.

(iii) The (DP) equation is a backward recursion in time (from which we get the optimum at $N-1$, then $N-2$ and so on.) The later policy is decided first.

It could also be instructive to remember the citation from Kierkegaard "*Life must be lived forward and understood backwards*".

3.4 Pontryagins Minimum Principle (PMP)

Another approach to finding an optimal controller that minimize an objective function is Pontryagins Minimum (or Maximum) Principle. It provides *necessary* conditions on the controller.

3.4.1 Discrete time

Consider the following discrete-time optimal control problem

$$\min \phi(x_N) + \sum_{k=0}^{N-1} f_0(k, x(k), u(k)) \quad \text{subject to} \begin{matrix} x(k+1) = f(k, x(k), u(k) \\ x_0 \text{ is given, } .G(x(N)) = 0 \end{matrix}$$

where $G(x) = (g_1(x), ..., g_p(x))^\top$ fulfils the usual regularity assumption, for example, the gradients $\{\nabla g_k(x)\}$ are linearly independent, and we take $\mathbb{R}^n$ and $\mathbb{R}^m$ as state space and control space, respectively.

The dynamic programming approach to solve such a control problem has the following properties.

- It produces feedback solutions, that is, we know the optimal control value for every position of the state vector x. This provides robustness to the closed loop system in the sense that ff the solution is perturbed by a disturbance then the controller still knows the optimal action.
- The solution is obtained by backwards iteration. It can be computationally demanding. One way to understand this is that we compute the optimal control value for every possible system state. What we win in robustness we loose in computational complexity.
- It is a sufficient condition.

Next we derive the Pontryagin's minimum principle to solve this problem. This can be done by standard optimization theory using Lagrange relaxation. In fact the PMP conditions are the first order necessary conditions, the so-called KKT conditions. To this end we recall the KKT conditions following Bazaraa et-al "Nonlinear programming": Suppose that x^* is a (local) optimum of

$$\min \mathcal{F}(x) \text{ subject to } h(x) = 0$$

where $\mathcal{F} : \mathbb{R}^n \to \mathbb{R}$ and $h : \mathbb{R}^n \to \mathbb{R}^p$ are continuously differentiable and the constraint set is regular, i.e., the gradients $\{\nabla h_k(x)\}$ are linearly independent (one of the constraints qualification condition). Then there is a vector $\lambda \in \mathbb{R}^p$ such that

1. $h(x^*) = 0$
2. $\nabla_x \mathcal{L}(x^*, \lambda) = 0$ where $\mathcal{L}(x^*, \lambda) = \mathcal{F}(x) + \lambda^\top h(x)$ is the *Lagrangian*, and the vector λ is the vector of Lagrange multipliers.

Theorem 3.4 (Pontryagins Minimum Principle (Discrete time))**.** Let $\{u^*(k)\}_{k=1}^{N-1}$ be an optimal control for above problem and let $\{x^*(k)\}_{k=0}^{N}$ be the corresponding trajectory. Then there exists an adjoint variable (Lagrange multiplier) $\{\lambda(k)\}_{k=1}^{N}$ such that

1. adjoint equation:

$$\lambda(k) = \frac{\partial H}{\partial x}(k, x^*(k), u^*(k), \lambda(k+1)),\ k = 1, ..., N-1$$

2. "pointwise optimization"

$$\frac{\partial H}{\partial u}(k, x^*(k), u^*(k), \lambda(k+1)) = 0,\ k = 0, 1, ..., N-1$$

3. boundary condition

$$\lambda(N) = \frac{\partial H}{\partial x}(x^*(N)) + h_x(x^*(N))^\top \nu$$

for some $\nu \in \mathbb{R}^p$.

where the Hamiltonian is

$$H(k, x, u, \lambda) = f_0(k, x, u) + \lambda^\top f(k, x, u)$$

Proof. Let $z^\top = \begin{pmatrix} x(1)^\top & \cdots & x(N)^\top & u(0)^\top & \cdots & u(N-1)^\top \end{pmatrix}$,

$$\mathcal{F}(z) = \phi(x_N) + \sum_{k=0}^{N-1} f_0(k, x(k), u(k))$$

$$h(z) = \begin{pmatrix} f(0, x(0), u(0) - x(1) \\ \vdots \\ f(N-1, x(N-1), u(N-1) - x(N) \\ G(x(N)) \end{pmatrix}$$

The KKT conditions (a necessary condition) for optimality of the problem

$$\min \mathcal{F}(x) \text{ subject to } h(x) = 0$$

are that there is a $\tilde{\lambda} = (\lambda^\top \ \nu^\top)^\top$ such that

$$\frac{\partial \mathcal{L}}{\partial z}(z^*, \tilde{\lambda}) = 0$$

where $\mathcal{L}(z, \tilde{\lambda}) = \mathcal{F}(z) + \tilde{\lambda}^\top h(z)$. More precisely

$$\frac{\partial \mathcal{L}}{\partial x(k)}(z^*) = \frac{\partial f_0}{\partial x}(k, x^*(k), u^*(k)) + \lambda(k+1)^\top \frac{\partial f}{\partial x}(k, x^*(k), u^*(k)) - \lambda(k), \ k = 1, ..., N-1$$

$$\frac{\partial \mathcal{L}}{\partial x(N)}(z^*) = \frac{\partial \phi}{\partial x}(x^*(N)) - \lambda(N) + G_x(x^*(N))^\top \nu$$

$$\frac{\partial \mathcal{L}}{\partial u}(z^*) = \frac{\partial f_0}{\partial u}(k, x^*(k), u^*(k)) + \lambda(k+1)^\top \frac{\partial f}{\partial u}(k, x^*(k), u^*(k)), \ k = 1, ..., N-1.$$

Thus, the condition $\frac{\partial \mathcal{L}}{\partial z}(z^*, \tilde{\lambda}) = 0$ together with the definition of the Hamiltonian H proves the theorem. □

This theorem is often used in the following way.

1. Define the Hamiltonian: $H(k, x, u, \lambda) = f_0(k, x, u) + \lambda^t opf(k, x, u)$.

2. Perform pointwise optmization, that is, find a function $\mu(k, x, \lambda)$ such that $\frac{\partial H}{\partial u}(k, x, u, \lambda) = 0$. Therefore the candidate optimal control is $u^*(k) = \mu(k, x^*(k), \lambda(k))$.

3. Solve the two boundary value problem

$$x(k+1) = \frac{\partial H}{\partial \lambda}(k, x(k), \mu(k, x(k), \lambda(k+1), \lambda(k+1)) = f(k, x(k), \mu(k, x(k), \lambda(k+1))),$$

$$\lambda(k) = \frac{\partial H}{\partial x}(k, x(k), \mu(k, x(k), \lambda(k+1), \lambda(k+1)), \lambda(N) = \frac{\partial \phi}{\partial x}(x(N)) + G_x(x(N))^\top \nu$$

with the boundary conditions $G(x(N)) = 0$ and $\lambda(N) = \frac{\partial \phi}{\partial x}(x(N)) + G_x(x(N))^\top \nu$

We call this a two point boundary value problem because the only unknown to determine are $\lambda(0)$ and $x(N)$. Once they are known all other state and adjoint variables can be computed from the recursive equations. It is interesting to note that the nonlinear program has a lot of structure that can be exploited.

The PMP approach is characterized by the following properties.

- It results in an open loop control problem, that is, the optimal solution is only known for a particular initial condition $x(0)$. If the solution is perturbed from the optimal by a disturbance then the optimal control may no longer be effective. The resulting system is therefore more sensitive to disturbances.

- It is generally easier to compute.
- It gives only a necessary condition for optimality.

There are few remarks in order. First this is an unconstrained optimal control problem. So it is easy to use the Lagrangian technique because we have equality constraints only, thus only two equations needed to solve in the KKT conditions. Let us consider the control space as a cube: $|u(i)| \leq 1$, $i = 0, ..., N-1$. Notice that these are functional inequality constraints. Then the KKT necessary condition are more involved. Instead we can move the constraints to the step where we pointwise optimize the Hamiltonian. In this particular example we can solve $\frac{\partial H}{\partial u}(k, x, u, \lambda) = 0$ together with the conditions $|u(i)| \leq 1$, $i = 0, ..., N-1$, or we use the KKT conditions to find the candidates of the optimization problem $\min\{H(k, x, u, \lambda) : |u(i)| \leq 1, \ i = 0, ..., N-1\}$. This is easier because we only have inequality constraints and the admissible set for u is convex and compact.

Next we know that the partial derivative with respect to u does not give us any thing if the Hamiltonian is linear in u. It is here the bang-bang control come into the picture. In the discrete setting it becomes a linear programming problem in this example, and we know that the optimal solution is the corner of the cube. So we choose either $u(i) = 1$ or $u(i) = -1$. It depends on the sign of the coefficient in case we only have one control variable.

3.4.2 Continuous time

Given an dynamical system

$$\dot{x} = f(x, u, t),$$

an objective function

$$J = Q(t_1) + \int_{t_0}^{t_1} q(x, u, t) dt$$

where the first term is the terminal cost, the second term is the trajectory cost, $t_0 \leq t \leq t_1$ where t_1 is a *free variable* , $x(t_1)$ is *specified* and u is *unconstrained*. We define the Hamiltonian

$$H = q(x, u, t) + pf(x, u, t)$$

where $p = [p_1, ..., p_n]$ is an Lagrange multiplier (or costate variable). Now the necessary conditions for an controller to be optimal is that

$$\frac{\partial H}{\partial u_i} = 0,$$

$$\dot{p_i} = -\frac{\partial H}{\partial x_i},$$

$$\Big(H + \frac{\partial Q}{\partial t}\Big)_{t=t_1}^{u=u^*} = 0,$$

$$\Big(H\Big)^{u=u^*} = 0 \quad \text{for } t_0 \leq t \leq t_1,$$

for $1 \leq i \leq n$.

These conditions are *necessary conditions* when $x(t_1)$ is specified, t_1 is a free variable and u is unconstrained. When the control problem differ from these constraint we will need other conditions.

We will now look at the case when u is constrained.

Theorem 3.5 (Pontryagins Minimum Principle (Continuous time))**.** *Given an objective function J and a dynamical system $\dot{x} = f(x, u, t)$ where $t_0 \leq t \leq t_1$ and t_1 is a free variable , $x(t_1)$ is specified and u is constrained. The necessary conditions on u^* to minimize J are*

$$H(x, u, p, t) \geq H(x^*, u^*, p^*, t).$$

$$\dot{p}_i = -\frac{\partial H}{\partial x_i},$$

$$\Big(H + \frac{\partial Q}{\partial t}\Big)_{t=t_1}^{u=u^*} = 0,$$

$$\Big(H\Big)^{u=u^*} = 0 \quad \textit{for } t_0 \leq t \leq t_1,$$

for $1 \leq i \leq n$.

where H is the Hamiltonian, and it is assumed that $x(t_1)$ is specfied and t_1 is free.

The *proof* is beyond the scope of this paper and thus omitted.

3.4.3 Example: Bang-Bang control

A landing vehicle separates from a spacecraft at time $t = 0$ at an altitude h from the surface of a planet, with initial downward velocity v. For simplicity assume that gravitational forces can be neglected and that the mass of the vehicle is constant. Consider vertical motion only, with upwards regarded as the positive direction. Let x_1 denote the altitude, x_2 velocity and $u(t)$ the thrust exerted by the rocket motor, subject to $|u(t)| \leq 1$ with suitable scaling. The equations of motion are

$$\dot{x}_1 = x_2, \quad \dot{x}_2 = u$$

with the initial conditions

$$x(0) = h, \quad x_2(0) = v.$$

In order to have a "soft landing" at some time t we require that

$$x_1(t_f) = 0, \quad x_2(t_f) = 0.$$

A suitable performance metric would be

$$\int_0^{t_f} (|u| + k)dt,$$

it represents a sum of total fuel consumption and time to landing, k being a factor which weights the relative importance of these two quantities.

To solve this problem we define the Hamiltonian

$$H = |u| + k + p_1 x_2 + p_2 u.$$

From PMP, the optimal controller must be of the following form

$$u^*(t) = \begin{cases} 1 & \text{if} \quad p_2^*(t) > 1 \\ 0 & \text{if} \quad 1 > p_2^*(t) > -1 \\ -1 & \text{if} \quad p_2^*(t) < -1 \end{cases}$$

Such a control is referred to in the literature by the graphic term bang-zero-bang, since only maximum thrust is applied in a forward or reverse direction; no intermediate nonzero values are used. If there is no period in which u^* is zero the control is called bang-bang. For example, a racing-car driver approximates to bang-bang operation, since he tends to use either full throttle or maximum braking when attempting to circuit a track as quickly as possible.

So the controller switches according to the value of $p_2^*(t)$, which is therefore termed (in this example) the switching function. One of the conditions were that

$$\dot{p_i} = -\frac{\partial H}{\partial x_i},$$

so we get that

$$\dot{p_1^*}(t) = 0, \quad \dot{p_2^*}(t) = -\dot{p_1^*}$$

and integrate

$$p_1^*(t) = c_1, \quad p_2^*(t) = c_1 t c_2,$$

where c_1, c_2 are constant. Since p_2^* is linear in t, it follows that it can take each of the values $+1$ and -1 at most once in$0 \leq t \leq tf$, so $u^*(t)$ can switch at most twice. We must however use physical considerations to determine an actual optimal control. Since the landing vehicle begins with a downwards velocity at an altitude h, logical sequences of control would seem to either

$$u^* = 0 \quad \text{followed by} \quad u^* = +1$$

(upwards is regarded as positive), or

$$u^* = -1, \quad \text{then} \quad u^* = 0, \quad \text{then} \quad u^* = +1.$$

Consider the first possibility and suppose that u^* switches from zero to one at time t_1. By virtue of of the controller this sequence of control is possible if p_2^*

decreases with time. It is easy to verify that the solution of $\dot{x}_1 = x_2, \quad \dot{x}_2 = u$ subject to the initial conditions $x(0) = h, \quad x_2(0) = v$ is

$$x_1^* = h - \mu t, \quad x_2^* = -v, \quad 0 \leq t \leq t_1$$

$$x_1^* = h - vt + \frac{1}{2}(t - t_1)^2, \quad x_2^* = -v + (t - t_1), \quad t_1 \leq t \leq t_f.$$

Substituting the soft landing requirements $x_1(t_f) = 0, x_2(t_f) = 0$ into the above gives

$$t_f = \frac{h}{v} + \frac{1}{2}v, t_1 = \frac{h}{v} - \frac{1}{2}v.$$

Because the final time is not specified and because of the form of the Hamiltonian H the equation $(H)_{u=u^*} = 0, t_0 \leq t \leq t_1$. holds, so in particular $(H)_{u=u^*}$ at $t = 0$, i.e. with $t = 0$ in H,

$$k = p_1^*(0)x_2^*(0) = 0$$

or $p_1^*(0) = k/v$. Hence from $p_1^*(t) = c_1, \quad p_2^*(t) = c_1 t c_2$ we have that

$$p_1^*(t) = k/v, t \geq 0$$

and

$$p_2^*(t) = -kt/v - 1 + kt_1/v$$

using the assumption that $p_2^*(t_1) = -1$. Thus the assumed optimal control will be valid if $t_1 > 0$ and $p_2^*(0) < 1$ (the latter conditions being necessary since $u^*(0) = 0$), and using $t_f = \frac{h}{v} - \frac{1}{2}v$ and $p_2^*(t) = -kt/v - 1 + kt_1/v$ these conditions imply

$$h > \frac{1}{2}v^2, \quad k < 2v^2/(h - 1/2v^2).$$

If these inequalities do not hold then some different control strategy, such as

$$u^* = -1, \quad \text{then} \quad u^* = 0, \quad \text{then} \quad u^* = +1,$$

becomes optimal. For example, if k is increased so that the inequality

$$k < 2v^2/(h - 1/2v^2)$$

is violated then this means that more emphasis is placed on the time to landing in the *performance index*. It is therefore reasonable to expect this time would be reduced by first accelerating downwards with $u^* = -1$ before coasting with $u^* = 0$, as in

$$u^* = -1, \quad \text{then} \quad u^* = 0, \quad \text{then} \quad u^* = +1.$$

It is interesting to note that provided

$$h > \frac{1}{2}v^2, \quad k < 2v^2/(h - 1/2v^2)$$

holds then the total time t_f to landing in

$$t_f = \frac{h}{v} + \frac{1}{2}v, t_1 = \frac{h}{v} - \frac{1}{2}v$$

is independent of k.

4 Reinforcement learning (RL)

Reinforcement learning (RL) is a branch of machine learning that involves learning a controller by sampling data from some environment. The goal of RL is learn optimal a controller by exploring the environment and adapting the controller based on the feedback it receives.

We follow the definition of RL and its problem formulations done in [3].

RL is most common defined through an Markov decision process (MDP), which is a discrete-time stochastic control process. The common notation for an MDP is that we have

- A set of states $X = \{x_1, x_2, ..., x_n\}$ that can be sampled from an environment, it is often called the state space.
- An set of controls (also called actions) $U = \{u_1, u_2, ..., u_m\}$ that represent all the inputs that are available each state. Depending on use-case, the controls can take different forms. But it is common that the controls have an magnitude and they are either single- or multidimensional.
- A transition function $x_{t+1} = f_t(x_t, u_t, w_t)$ that maps probability of transitioning to a certain state given some control and current state since the the transition function is stochastic. Note that w_t is the state transition randomness.
- A cost function $c(x_t, u_t)$, which maps a state-controls pair to a scalar cost.
- A discount factor $\gamma \in [0, 1]$, which is often used to penalize cycles of states.

Over the course of T sampling steps, we will have generated a pairs of values $(u_0, x_0, x_1), ..., (u_{T-1}, x_{T-1}, x_T)$ and associated costs $c_0, ..., c_T$. The goal of RL is to minimize the cumulative cost

$$V^\pi = E\left[\sum_{t=0}^{T} \gamma^t c(u_t, x_t, x_{t+1}) + c_T(x_T)\right]$$

given a control policy

$$\pi = \{\pi_0, \pi_1, ..., \pi_{T-1}\}$$

where $u_t = \pi_t(u_{t-1}, x_t)$.

To minimize V we need to find an control policy that chooses controls u_t that minimize V. For a given state x_t the policy π_t maps a probability that the control u_t will minimize V.

The value function $V_\pi(x)$ is defined as the expected cumulative reward that can be obtained by following the control policy π from a given state x_0. The function is used to evaluate different policies.

The problem of RL is that the optimal policy function that minimizes (or maximizes) the value function is often not known to us, therefore the problem is to find such policy.

Remark: Some definitions of RL have instead the objective of maximizing the value function. In this case you often speak of *rewards* instead of *costs.*

How the optimal policy is found, depends on different factors. Such as whether the dynamical system have *known or unknown dynamics and cost*, the *computational resources* that are available and *dimensionality and cardinality* of the system, the number of *boundary condition or initial states.* This leads us to different kinds of problems.

If either the dynamics or cost are unknown we will have to learn it in some way, this leads us to two approaches. The first one is multi(or episodic)-trajectory, meaning that after T number of steps in the trajectory we stop and reset, then start from another (or same) initial state. After some number of trajectories have been sampled, we approximate the dynamics from the gathered data. In contrast a single-trajectory setting we only consider one trajectory and the dynamics will have to be approximated only through this trajectory. So in a singe-trajectory there is only one initial state, while in a multi-trajectory settings there are multiple initial states that are used. A more classic name for single- and multi-trajectory settings is respectively adaptive learning control and iterative learning control.

Even when the dynamics and cost is known our *computational resources* may not be enough to compute an exact solution due to high *dimensionality and cardinality* of the problem. In this case, approximation methods can be used instead, but the solution might not be exact.

As we might have seen RL and Control Theory share the common goal of optimizing system behavior to achieve desired objectives. The main distinction between RL and optimal control theory is that RL mostly deals with finding a control policy by *learning from data*, while control theory most often deals with *deriving* an optimal control policy from a known mathematical model of a system.

In RL, there is a distinction between model-free and model-based approaches. Model-free RL methods learn from experience without explicitly modeling the environment, relying on data to approximate the system dynamics and costs. Model-based RL methods, on the other hand, create explicit models of the system and use them for finding an optimal policy.

Control Theory mainly relies on explicitly known mathematical models of the dynamical system. The control design is based on these known models, and mathematical tools like differential- or difference equations.

4.1 Known Dynamics and cost: Dynamic programming

When the dynamics are stochastic, a model of the system will be

$$x(t+1) = Ax(t) + Bu(t) + w(t)$$

where $w(t) \overset{i.i.d}{\sim} \mathcal{N}(0, \sigma_w^2 I)$.

In this case, it is still possible to derive a solution given that we have been given or defined a cost function. Using dynamic programming is one approach to finding the optimal controller. We will closely study the LQR problem which is a classical instantiation of the MDP, with the help of Dynamic programming.

Example: Stochastic Linear Quadratic Regulator (SLQR)

The LQR problem is often studied when the objective is to keep the system close to the origin. The goal is to minimize the following cost function

$$J = E\big[\sum_{i=0}^{T-1}(x_i^\top Q x_i + u_i^\top R u_i) + x_T^\top Q x_T\big],$$

with respect to the controls $u_0, ..., u_{N-1}$ and where R is positive definite matrix and Q is a positive semi-definite matrix. The solution is based up lecture notes from [4].

We will solve this problem via Dynamic programming. So, we will solve this inductively backwards starting form the final state x_T.

We define the value function as

$$V(t, x_t) := \min_{u_t, u_{t+1}, \dots, u_{T-1}} \sum_{i=t}^{T-1} E\big[x_i^\top Q x_i + u_i^\top R u_i\big] + x_T^\top Q x_T$$

The cost at x_T is $V(T, x_T) = x_T^\top Q x_T$. Now we solve for $V(t, x_t)$ in terms of the next step $V(t+1, x_{t+1})$. We have that

$$V(t, x_t) = \min_{u_t, u_{t+1}, \dots, u_{T-1}} \sum_{i=t}^{T-1} E\big[x_i^\top Q x_i + u_i^\top R u_i + x_T^\top Q x_T\big].$$

we pull the first step out of the sum

$$\min_{u_t} x_t^\top Q x_t + u_t^\top R u_t + \big(\min_{u_{t+1}, \dots, u_{T-1}} \sum_{i=t+1}^{T-1} E\big[x_i^\top Q x_i + u_i^\top R u_i + x_T^\top Q x_T\big]\big) \Leftrightarrow$$

$$\Leftrightarrow V(t, x_t) = \min_{u_t} x_t^\top Q x_t + u_t^\top R u_t + E\big[V(t+1, x_{t+1})\big] \Leftrightarrow$$

$$\Leftrightarrow V(t,x_t) = \min_{u_t} x_t^\top Q x_t + u_t^\top R u_t + E\big[V(t+1, Ax(t)+Bu(t)+w(t))\big].$$

We make an ansatz $V(t,x_t) = x_t^\top P_t x_t + r_t$ where $P \in R^{n\times n}$ is a semi-definite matrix, r_t is a constant and $r_T = 0$. It is true for T since $V(T,x_T) = x_T^\top Q x_T + 0$ and Q is a semi-definite matrix. Now assume it is true for $k+1$ for

$$V(t,x_t) = \min_{u_t} x_t^\top Q x_t + u_t^\top R u_t + E\big[V(t+1, Ax(t)+Bu(t)+w(t))\big].$$

We expand the third term

$$\begin{gathered} E\big[V(t+1, Ax(t)+Bu(t)+w(t))\big] = \\ = E\big[(Ax(t)+Bu(t)+w(t))^\top P_{k+1}(Ax(t)+Bu(t)+w(t)) + r_{t+1}\big] = \\ = E\big[(Ax(t)+Bu(t)+w(t))^\top (P_{t+1}Ax(t)+P_{t+1}Bu(t)+P_{t+1}w(t)) + r_{t+1}\big] = \\ = E\big[x(t)^\top A^\top (P_{t+1}Ax(t)+P_{t+1}Bu(t)+P_{t+1}w(t)) + \\ + u(t)^\top B^\top (P_{t+1}Ax(t)+P_{t+1}Bu(t)+P_{t+1}w(t)) + \\ + w(t)^\top (P_{t+1}Ax(t)+P_{t+1}Bu(t)+P_{t+1}w(t)) + r_{t+1}\big]. \end{gathered}$$

Now since $E[w(t)] = 0$ and $E[\alpha w(t)] = \alpha E[w(t)]$ for some constant $\alpha \in \mathbb{R}$ we have that

$$\begin{gathered} x_t^\top Q x_t + u_t^\top R u_t + x(t)^\top A^\top P_{t+1}Ax(t) + x(t)^\top A^\top P_{t+1}Bu(t) + \\ + u(t)^\top B^\top P_{t+1}Ax(t) + u(t)^\top B^\top P_{t+1}Bu(t) + E[w(t)^\top P_{t+1}w(t)] + r_{t+1} = \end{gathered}$$

$$\begin{gathered} = x_t^\top (A^\top P_{t+1}Ax(t) + Q)x_t + u_t^\top (B^\top P_{t+1}Bu(t) + R)u_t + \\ + x(t)^\top A^\top P_{t+1}Bu(t) + u(t)^\top B^\top P_{t+1}Ax(t) + E[w(t)^\top P_{t+1}w(t)] + r_{t+1}. \end{gathered}$$

Since $w(t) \overset{i.i.d}{\sim} \mathcal{N}(0, \sigma_w^2 I)$, the covariance between the components of $w(t)$ is equal to zero, we have that

$$E[w(t)^\top P_{t+1} w(t)] = E[\sum_i^n \sum_j^n w_i P_{ij} w_j] = \sum_i^n \sum_j^n E[w_i w_j] P_{ij} =$$

$$= \sum_i^n \sum_j^n \sigma_w^2 I_{ij} P_{ij} = Tr(\sigma_w^2 P_{t+1}).$$

Now we want to find the optimal controller

$$\begin{gathered} \arg\min_u \quad x_t^\top Q x_t + u_t^\top R u_t + x(t)^\top A^\top P_{t+1}Ax(t) + x(t)^\top A^\top P_{t+1}Bu(t) + \\ + u(t)^\top B^\top P_{t+1}Ax(t) + u(t)^\top B^\top P_{t+1}Bu(t) + Tr(\sigma_w^2 P_{t+1}) + r_{t+1}. \end{gathered}$$

According to the matrix calculus from our Linear Algebra preliminaries

$$\frac{\partial(x^\top Ax)}{\partial x} = x^\top(A + A^\top), \quad \frac{\partial(y^\top Ax)}{\partial x} = y^\top A, \quad \frac{\partial(x^\top Ay)}{\partial x} = y^\top A^\top$$

for $A \in \mathbb{R}^{n\times n}$ and $x, y \in \mathbb{R}^n$.

Derivate the expression with respect to u to find where the gradient is equal to zero, so the optimal controller can be found

$$x(t)^\top AP_{t+1}B + x(t)^\top AP_{t+1}B + 2u(t)^\top BP_{t+1}B + 2u(t)^\top R = 0 \Leftrightarrow$$

$$\Leftrightarrow 2x(t)^\top A^\top P_{t+1}B + 2u(t)^\top B^\top P_{t+1}B2u(t)^\top R = 0$$

$$\Leftrightarrow u(t)^\top B^\top P_{t+1}B + u(t)^\top R = -x(t)^\top A^\top P_{t+1}B$$

$$\Leftrightarrow (B^\top P_{t+1}B + R)u(t) = -B^\top P_{t+1}Ax(t) \Leftrightarrow$$

$$\Leftrightarrow u^*(t) = -(B^\top P_{t+1}B + R)^{-1}B^\top P_{t+1}Ax(t).$$

Now if the optimal controller u^* is applied to

$$x_t^\top(A^\top P_{t+1}A + Q)x_t + u_t^\top(B^\top P_{t+1}B + R)u_t +$$

$$+x(t)^\top A^\top P_{t+1}Bu(t) + u(t)^\top B^\top P_{t+1}Ax(t) + Tr(\sigma_w^2 P_{t+1}) + r_{t+1}$$

the second term and fourth term will cancel out each other and we are left with

$$x_t^\top(A^\top P_{t+1}A + Q)x_t + Tr(\sigma_w^2 P_{t+1}) -$$

$$-x(t)^\top A^\top P_{t+1}B(B^\top P_{t+1}B + R)^{-1}B^\top P_{t+1}Ax(t) + r_{t+1} =$$

$$= x_t^\top\Big[A^\top P_{t+1}A + Q - A^\top P_{t+1}B(B^\top P_{t+1}B + R)^{-1}B^\top P_{t+1}A\Big]x_t +$$

$$+Tr(\sigma_w^2 P_{t+1}) + r_{t+1}.$$

Now, from our Linear algebra Preliminaries

1. The inverse of a positive definite matrix $P \in \mathbb{R}^{n\times n}$ is also a positive definite matrix

2. If $P \in \mathbb{R}^{n\times n}$ is a positive definite matrix then it is invertible.

3. If $P \in \mathbb{R}^{n\times n}$ is a positive definite matrix and $Q \in \mathbb{R}^{n\times n}$ is a semi-definite matrix then $P + Q$ is positive definite matrix.

4. If $P \in \mathbb{R}^{n\times n}$ is a positive definite matrix and $A \in \mathbb{R}^{n\times n}$. Then $A^\top PA$ is a positive semi-definite matrix. If A is invertible, then $A^\top PA$ is a positive definite matrix.

Our induction hypothesis was that $V(t,x) = x_t^\top P_t x_t + r_t$. Now applying the rules above, the terms $A^\top P_{t+1} A + Q$ and $A^\top P_{t+1} B(B^\top P_{t+1} B + R)^{-1} B^\top P_{t+1} A$ are clearly positive semi-definite matrices. Therefore our induction hypothesis holds. In the process we also found the optimal controller.

To summarize we have proved that $V(t,x_t) = x^\top P_t x + r_t$ where P is semi-definite matrix and r_t is constant, and

$$P_T = Q, \quad r_T = 0$$

$$P_t = A^\top P_{t+1} A + Q - A^\top P_{t+1} B(B^\top P_{t+1} B + R)^{-1} B^\top P_{t+1} A$$

$$r_t = Tr(\sigma_w^2 P_{t+1}) + r_{t+1}$$

$$u_t^* = -(B^\top P_{t+1} B + R)^{-1} B^\top P_{t+1} A x(t).$$

Given any initial state x_0 and some final time T the cost of the LQR will be

$$V(0,x_0) = x_0^t P_0 x_0 + r_0 = x_0^t P_0 x_0 + \sum_{t=0}^{T} Tr(P_t \sigma_w^2),$$

and again the matrix P_0 is calculated recursively backwards in time starting from the final time. For an example if we want to compute P_{T-1}, we know that $P_T = Q$. We simply use the formula above for P_t, and we get

$$P_{T-1} = A^\top Q A + Q - A^\top Q B(B^\top Q B + R)^{-1} B^\top Q A.$$

After this P_{T-2} can be obtained through P_{T-1} and so on.

4.2 Unknown Dynamics: Model-free vs Model-Based methods

In the Stochastic Linear Quadratic Regulator problem we could derive an optimal controller since the dynamics were *known*. But finding the optimal policy becomes particularly challenging when the underlying dynamics of the system are *unknown*. In such cases, two fundamental approaches emerge: *model-based* and *model-free* methods. Model-based methods strive to construct an explicit model of the system's dynamics. In other words, we try to approximate the matrices A and B in a dynamical system. Then using the model to determine the optimal policy. On the other hand, model-free methods forgo the explicit modeling of the dynamics, instead focusing on learning a policy directly from the interactions with the environment. Both approaches offer distinct advantages and trade-offs. Forward we will choose the control theoretic approach which is the model-based method. This will lead us to the subject of *System identification.*

5 System Identification

In this section we will show how to construct a model of an unknown discrete dynamical system with *multiple-trajectories*. But we will disregard the whether the cost function is *known or unknown* in this section. Then we will analyze how a Bang-Bang controller will affect the error of the approximation of the system. We will under reasonable assumptions show that the Bang-Bang controller provides a better approximation of a discrete dynamical system.

The following derivation of the error terms is based from [2].

5.1 Upper bound and errors

In the case when the system is unknown and we chose a model-based approach to find an optimal controller, we will estimate A and B. We will assume that the system is controllable this implies that that the Controllability Gramian

$$\Lambda_C(A, B, T) = \sum_{i=0}^{T} A^i B B^\top (A^i)^\top$$

is positive definite matrix for some T.

5.1.1 Derive an error term

Consider the stochastic discrete dynamical system

$$x(t+1) = Ax(t) + bu(t) + w(t),$$

where $x_t, w_t \in \mathbb{R}^n, u \in \mathbb{R}^m$ where $w_t \overset{i.i.d}{\sim} \mathcal{N}(0, \sigma_w^2 I)$. We also inject noise through $u \overset{i.i.d}{\sim} \mathcal{N}(0, \sigma_u^2 I)$. To simplify the analysis, we will use only use the last two samples at $T+1, T$ to analyze the error estimates. We sample N trajectories with $T+1$ time-steps from the dynamical system. The goal is to find an estimation of A, B such that

$$\sum_{i=1}^{N} ||x_{T+1}^{(i)} - Ax_T^{(i)} - Bu_T^{(i)}||_2^2.$$

is minimized.

To find the optimal estimates of A, B we derivate the expression to search where the derivate is equal to zero. We have that

$$\nabla_{(A,B)} \sum_{i=1}^{N} ||x_{T+1}^{(i)} - Ax_T^{(i)} - Bu_T^{(i)}||_2^2 = 0.$$

We notice that for some i that

$$||x_{T+1}^{(i)} - Ax_T^{(i)} - Bu_T^{(i)}||_2^2 = \sum_{k=1}^{n} \Big[(x_{T+1}^{(i)})_k - (Ax_T^{(i)})_k - (Bu_T^{(i)})_k\Big]^2.$$

If we let

$$X_N := \begin{bmatrix} (x_{T+1}^{(1)})^\top \\ \vdots \\ (x_{T+1}^{(N)})^\top \end{bmatrix}, Z_N := \begin{bmatrix} (x_T^{(1)})^\top, & (u_T^{(1)})^\top \\ \vdots & \vdots \\ (x_T^{(N)})^\top, & (u_T^{(N)})^\top \end{bmatrix}, W_N := \begin{bmatrix} (w_T^{(1)})^\top \\ \vdots \\ (w_T^{(N)})^\top \end{bmatrix}.$$

then the expression can we rewritten as

$$\nabla_{(A,B)} \sum_{i=1}^{N} ||x_{T+1}^{(i)} - Ax_T^{(i)} - Bu_T^{(i)}||_2^2 =$$

$$= \nabla_{(A,B)} ||X_N - Z_N \big[A \ \ B\big]^\top||_F^2 =$$

$$= \nabla_{(A,B)} tr\big[\big(X_N - Z_N\big[A \ \ B\big]^\top\big)\big(X_N - Z_N\big[A \ \ B\big]^\top\big)^\top\big] =$$

$$= \nabla_{(A,B)} tr\big[X_N X_N^\top - X_N\big[A \ \ B\big]Z_N^\top - Z_N\big[A \ \ B\big]^\top X_N^\top + Z_N\big[A \ \ B\big]^\top\big[A \ \ B\big]Z_N^\top\big].$$

Now we derivate the expression with respect to the concatenated matrix (A, B). We use the propositions (2.8) and (2.12) about derivatives of *Trace* from our Linear Algebra preliminaries and we get that

$$-2X_N^\top Z_N + 2\big[A \ \ B\big]Z_N^\top Z_N = 0.$$

We apply the pseudo-inverse of Z_N and $Z_N^\top$,

$$-X_N^\top + \big[A \ \ B\big]Z_N^\top = 0 \Leftrightarrow$$

$$\Leftrightarrow Z_N\big[A \ \ B\big]^\top = X_N \Leftrightarrow$$

$$\Leftrightarrow \big[A \ \ B\big]^\top = (Z_N^\top Z_N)^{-1} Z_N^\top X_N \Leftrightarrow$$

$$\Leftrightarrow \big[A \ \ B\big]^\top = (Z_N^\top Z_N)^{-1} Z_N^\top (Z_N\big[A \ \ B\big]^\top + W_N) \Leftrightarrow$$

$$\Leftrightarrow \big[A \ \ B\big]^\top = \big[A \ \ B\big]^\top + (Z_N^\top Z_N)^{-1} Z_N^\top W_N.$$

We have now derived an error term for the estimate

$$[\hat{A} \ \ \hat{B}\big]^\top = \big[A \ \ B\big]^\top + (Z_N^\top Z_N)^{-1} Z_N^\top W_N.$$

In the derivation we also found a formula for computing the estimates $\hat{A}, \hat{B}$, namely

$$\big[\hat{A} \ \ \hat{B}\big]^\top = (Z_N^\top Z_N)^{-1} Z_N^\top X_N.$$

5.1.2 Derive the Covariance matrix

What is the formula for state at $x(k+1)$? Assume we start at 0, i.e. $x(0) = 0$, then we have that

$$x(1) = Bu(0) + w(0)$$

$$x(2) = A(Bu(0) + w(0)) + Bu(1) + w(1) = ABu(0) + Aw(0) + Bu(1) + w(1)$$

$$x(3) = A(ABu(0) + Aw(0) + Bu(1) + w(1)) + Bu(2) + w(2) =$$

$$= A^2Bu(0) + A^2w(0) + ABu(1) + Aw(1) + Bu(2) + w(2)$$

$$\vdots$$

We guess that the formula for $x(k+1)$ is

$$x(k+1) = \sum_{i=0}^{k} A^iBu(k-i) + A^iw(k-i).$$

The formula is true for $x(1), x(2)$. We assume it is true for $k+2$,

$$A\Big[\sum_{i=0}^{k} A^iBu(k-i) + A^iw(k-i)\Big] + Bu(k+1) + w(k+1) =$$

$$= \Big[\sum_{i=0}^{k} A^{i+1}Bu(k-i) + A^{i+1}w(k-i)\Big] + Bu(k+1) + w(k+1) =$$

$$= \sum_{i=0}^{k+1} A^iBu(k+1-i) + A^iw(k+1-i) = x(k+2).$$

The formula holds.

Since $x(k+1)$ is a random vector, we want to understands its variance. We compute its variance

$$Var(x(k+1)) = Cov(x(k+1), x(k+1)^\top) =$$

$$E\Big[\big[\sum_{i=0}^{k} A^iBu(k-i) + A^iw(k-i) - 0\big]\big[\sum_{i=0}^{k} A^iBu(k-i) + A^iw(k-i) - 0\big]^\top\Big] =$$

$$E\Big[\sum_{i=0}^{k} A^iBu(k-i)u(k-i)^\top B^\top (A^i)^\top + A^iw(k-i)w(k-i)^\top (A^i)^\top + ...\Big].$$

We recall that the $E[aX] = aE[X]$ and if two random variables are independent then $E[XY] = E[X]E[Y]$. Since $u(0), ..., u(k), w(0), ..., w(k-i)$ are independent variables, the remaining mixed terms of $Var(x(k+1))$ will be equal to zero because the expected value of $u(0), ..., u(k), w(0), ..., w(k-i)$ is zero. So,

$$E\Big[\sum_{i=0}^{k} A^i Bu(k-i)u(k-i)^\top B^\top (A^i)^\top + A^i w(k-i)w(k-i)^\top (A^i)^\top\Big] =$$

$$= E\Big[\sum_{i=0}^{k} A^i Bu(k-i)u(k-i)^\top B^\top (A^i)^\top\Big] + E\Big[A^i w(k-i)w(k-i)^\top (A^i)^\top\Big] =$$

$$= \sigma_u^2 \sum_{i=0}^{k} A^i BB^\top (A^i)^\top + \sigma_w^2 \sum_{i=0}^{k} A^i (A^i)^\top = Var(x(k+1)).$$

Now it is easy verifiable that

$$\begin{bmatrix} x_T^{(i)} \\ u_T^{(i)} \end{bmatrix} \overset{i.i.d}{\sim} \mathcal{N}\Bigg(0, \begin{bmatrix} \sigma_u^2 \Lambda_C(A,B,T) + \sigma_w^2 \Lambda_C(A,I,T) & 0 \\ 0 & \sigma_u^2 I_{n_u} \end{bmatrix}\Bigg),$$

where n_u is the dimension of u. We define Σ_x as the first block of the above covariance matrix

$$\Sigma_x = \sigma_u^2 \sum_{i=0}^{k} A^i BB^\top (A^i)^\top + \sigma_w^2 \sum_{i=0}^{k} A^i (A^i)^\top,$$

and Σ_u as the second block

$$\Sigma_u = \sigma_w^2 I_{n_u}.$$

The first block of covariance matrix for the Gaussian and Bang-bang controller will be denoted Σ_{x_G} and respectively Σ_{x_B}.

5.1.3 Derive upper bound on error term A

We will now derive a probability bounds with the spectral norm of the error terms. To remind, if $A \in \mathbb{R}^{n\times n}$ and singular values of A is $\sigma_1 \leq ... \leq \sigma_n$, the spectral norm is defined as

$$||A||_2 = \max_{||x||_2=1} ||Ax||_2 = \sigma_n$$

in other words it is the maximum singular value of the matrix A, this was shown in our Linear Algebra preliminaries.

From the derived error terms from earlier

$$\begin{bmatrix}\hat{A} & \hat{B}\end{bmatrix}^\top = \begin{bmatrix}A & B\end{bmatrix}^\top + (Z_N Z_N^\top)^{-1} Z_N^\top W_N,$$

we can easily verify that the error terms for A and B is

$$\begin{bmatrix}\hat{A} - A\end{bmatrix}^\top = [I_{n_x} \quad 0_{n_x \times x_u}](Z_N^\top Z_N)^{-1} Z_N^\top W_N,$$

$$\begin{bmatrix}\hat{B} - B\end{bmatrix}^\top = [0_{n_u \times x_x} I_{n_u}](Z_N^\top Z_N)^{-1} Z_N^\top W_N,$$

where n_x, n_u is the dimension of x and respectively u.

Define $Q_A = [I_{n_x} \quad 0_{n_x \times x_u}]$, the the spectral norm of the error term is

$$||\hat{A} - A||_2 = ||Q_A(Z_N^\top Z_N)^{-1} Z_N^\top W_N||_2.$$

For the matrix Z_N the expected distance is the square root of the covariance matrix and the spectral norm is a distance preserving. Therefore, Z_N can rewrite to $Z_N = Y_N \Sigma^{1/2}$ where

$$Y_N := \begin{bmatrix} y_1^\top \\ \vdots \\ y_N^\top \end{bmatrix}$$

with $y_i \overset{i.i.d}{\sim} \mathcal{N}(0, I_{n_x+n_u})$. We have that

$$||Q_A(Z_N^\top Z_N)^{-1} Z_N^\top W_N||_2 = ||Q_A((Y_N\Sigma^{1/2})^\top Y_N \Sigma^{1/2})^{-1}(Y_N\Sigma^{1/2})^\top W_N||_2 =$$

$$= ||Q_A(\Sigma^{1/2} Y_N^\top Y_N \Sigma^{1/2})^{-1} \Sigma^{1/2} Y_N^\top W_N||_2 =$$
$$= ||Q_A \Sigma^{-1/2} (Y_N^\top Y_N)^{-1} Y_N^\top W_N||_2.$$

Assuming the singular values of M is $\sigma_1 \leq ... \leq \sigma_n$, the spectral norm for the inverted matrix M is

$$||M^{-1}||_2 = \frac{1}{\sigma_1}$$

which was proved in our Linear algebra preliminaries.

Since the we assumed that the dynamical system were controllable, Σ_x will be an positive definite matrix because. Therefore its eigenvalues are equal to its singular values. From our Linear Algebra preliminaries, we proved that the Spectral norm is submultiplicative. Now we have that

$$= ||Q_A \Sigma^{-1/2} (Y_N^\top Y_N)^{-1} Y_N^\top W_N||_2 \leq$$

$$\leq ||\Sigma_x^{-1/2}||_2 \frac{||Y_N^\top W_N||_2}{||Y_N^\top Y_N||_2} = \lambda_{min}(\Sigma_x^{-1/2}) \frac{||Y_N^\top W_N||_2}{\lambda_{min}(Y_N^\top Y_N)}.$$

To conclude,

$$||\hat{A} - A||_2 \leq \lambda_{min}(\Sigma_x^{-1/2}) \frac{||Y_N^\top W_N||_2}{\lambda_{min}(Y_N^\top Y_N)}.$$

and, in the paper [2] the researchers prove that given some parameters, the term

$$\frac{||Y_N^\top W_N||_2}{\lambda_{min}(Y_N^\top Y_N)}$$

has an upper bound. The *proof* is outside the scope of this paper thus omitted.

5.1.4 Derive upper bound on error term B

To derive an upper bound for the the error term for B we will use the same arguments that were done in derivation for the error term for A.

We have that

$$\begin{aligned}||\hat{B} - B||_2 = ||Q_B(Z_N^\top Z_N)^{-1} Z_N^\top W_N||_2 = \\ = ||Q_B \Sigma^{-1/2}(Y_N^\top Y_N)^{-1} Y_N^\top W_N||_2 \leq \\ \leq ||(\sigma_u^2 I_{n_u})^{-1/2}||_2 \cdot \frac{||Y_N^\top W_N||_2}{\lambda_{min}(Y_N^\top Y_N)} = \\ = \frac{1}{\sigma_u} \frac{||Y_N^\top W_N||_2}{\lambda_{min}(Y_N^\top Y_N)}.\end{aligned}$$

5.2 Error Bounds for Bang-Bang- VS Gaussian controller

We have concluded that the error bounds for A respectively B is

$$||\hat{A} - A||_2 \leq \lambda_{min}(\Sigma_x^{-1/2}) \frac{||Y_N^\top W_N||_2}{\lambda_{min}(Y_N^\top Y_N)}$$

$$||\hat{B} - B||_2 \leq \frac{1}{\sigma_u} \frac{||Y_N^\top W_N||_2}{\lambda_{min}(Y_N^\top Y_N)}.$$

We now want to understand how using a Bang-bang controller compared to a Gaussian controller affects the error terms. In many control problems, our controller u is often limited by some magnitude $||u|| \leq U$, where U is some scalar. In the case where $u \in \mathbb{R}^m$ is a vector, each component of the controller u will have the following form

$$u_i = \begin{cases} U & : Pr(X = U) = 0.5 \\ -U & : Pr(X = -U) = 0.5 \end{cases}$$

for $1 \leq i \leq m$.

5.2.1 Standard deviation for the controllers

The components of the Gaussian controller will approximately be $\mathcal{N}(0, 0.5^2)$ multiplied by the maximum magnitude U of the controller.

For the variance for each of the Bang-Bang controller will be

$$Var(X_B) = 0.5 \cdot (U - 0)^2 + 0.5 \cdot (-U - 0)^2 = U^2$$

so the the standard deviation for the Bang-Bang controller will be U. While the standard deviation for the Gaussian controller will be $0.5 \cdot U$.

A direct observation for this is that when using the Bang-bang controller, the error bound for B which is

$$\frac{1}{\sigma_u} \frac{||Y_N^\top W_N||_2}{\lambda_{min}(Y_N^\top Y_N)}$$

will be halved compared to using an Gaussian controller.

5.2.2 Smallest eigenvalues of Σ_x with different controllers

For the error bound of A we will analyze the term $\lambda_{min}(\Sigma_x)$ to show that when using Bang-bang controller, this error bound also get reduced compared to using a Gaussian controller. To prove this we will use the Rayleigh quotient.

Firstly, by proposition (2.7) the matrix $\Lambda_C(A, I, T)$ is semi-definite matrix if A is not invertible and if A is invertible it is positive definite.

Since we assumed that the system is controllable, $\sigma_u^2 \Lambda_C(A, B, T)$ is a positive definite matrix. Therefore by proposition (2.4) the matrix $\sigma_u^2 \Lambda_C(A, B, T) + \sigma_w^2 \Lambda_C(A, I, T)$ is a positive definite matrix.

This implies that all the eigenvalues of Σ_x is positive and bigger then zero. Assume we have have (invertible) symmetric matrix $M \in \mathbb{R}^{n \times n}$. To remind the Rayleigh quotient of a symmetric matrix is defined as

$$R(M, x) = \frac{x^t M x}{x^t x}.$$

Assume that the eigenvalues of $M \in \mathbb{R}^{n \times n}$ is $0 < \lambda_1 \leq ... \leq \lambda_n$. From the Courant–Fischer–Weyl min-max principle (Theorem 2.1) we have that the eigenvalues of M is

$$\lambda_1(M) \leq R(M, x) \leq \lambda_n(M)$$

because

$$\min_x R(M, x) = \lambda_1(M), \quad \max_x R(M, x) = \lambda_n(M).$$

We denote $M_1 = \sigma_u^2 \Lambda_C(A, B, T)$ and $M_2 = \sigma_w^2 \Lambda_C(A, I, T)$. Now consider

$$\lambda_1(M_2) \leq R(M_2, x) \leq \lambda_n(M_2),$$

if we now add

$$\lambda_1(M_1) = \min_x R(M_1, x)$$

to the equation we get that

$$\lambda_1(M_2) + \lambda_1(M_1) \leq R(M_2, x) + \lambda_1(M_1).$$

Now $\lambda_1(M_2) + \lambda_1(M_1) \leq R(M_1 + M_2, x)$ must be true since

$$\min_{x_1} \frac{x_1^t M_1 x_1}{x_1^t x_1} + \min_{x_2} \frac{x_2^t M_2 x_2}{x_2^t x_2} \leq \frac{x^t (M_1 + M_2) x}{x^t x} = \frac{x^t M_1 x}{x^t x} + \frac{x^t M_2 x}{x^t x}$$

for any $x \in \mathbb{R}$, therefore

$$\lambda_1(M_1) + \lambda_1(M_2) \leq \lambda_1(M_1 + M_2).$$

Now if we instead have a positive scalar $a \in \mathbb{R}^+$ in front of M_1, how will this affect the minimum value of $aM_1 + M_2$? We examine the difference between $R(aM_1 + M_2, x)$ and $R(M_1 + M_2, x)$. Assume $R(aM_1 + M_2, x) > R(M_1 + M_2, x)$, now

$R(aM_1 + M_2, x) > R(M_1 + M_2, x) \Leftrightarrow R(aM_1 + M_2, x) - R(M_1 + M_2, x) > 0 \Leftrightarrow$

$$\Leftrightarrow \frac{x^t(aM_1 + M_2 - M_1 + M_2)x}{x^t x} = \frac{(a-1)x^t M_1 x}{x^t x} > 0.$$

Now if $a \geq 2$ then

$$\frac{(a-1)x^t M_1 x}{x^t x} > 0$$

holds.

So the smallest difference between $R(M_1 + M_2, x)$ and $R(aM_1 + M_2, x)$ is $(a-1)\lambda_1(M_1)$ since

$$\min_x R((a-1)M_1, x) = (a-1)\lambda_1(M_1).$$

In our case with the Gaussian controller

$$\Sigma_{x_G} = \sigma_u^2 \Lambda_C(A, B, T) + \sigma_w^2 \Lambda_C(A, I, T)$$

with the Bang-Bang controller

$$\Sigma_{x_B} = (2 \cdot \sigma_u)^2 \Lambda_C(A, B, T) + \sigma_w^2 \Lambda_C(A, I, T) = 4 \cdot \sigma_u^2 \Lambda_C(A, B, T) + \sigma_w^2 \Lambda_C(A, I, T).$$

So, using the formula above $\frac{(a-1)x^t M_1 x}{x^t x} > 0$, in our case $a = 4$, so the minimum eigenvalue of Σ_{x_B} will at least differ by $3\lambda_1(\Lambda_C(A, B, T))$ then the minimum eigenvalue of Σ_{x_G}.

Therefore

$$\frac{1}{\lambda_1(\Sigma_{x_B}^{1/2})} < \frac{1}{\lambda_1(\Sigma_{x_G}^{1/2}) + 3\lambda_1(\Lambda_C(A, B, T))} \leq \frac{1}{\lambda_1(\Sigma_{x_G}^{1/2})}$$

so the error

$$||\hat{A} - A||_2 \leq \lambda_{min}(\Sigma_x^{-1/2}) \frac{||Y_N^\top W_N||_2}{\lambda_{min}(Y_N^\top Y_N)}$$

will be less when using the Bang-Bang controller compared to the Gaussian controller.

6 Conclusions

In RL problems where model-based methods are being used, we have now showed that the approximation of the environment will be better when using Bang-bang controllers instead of a Gaussian controller.

We have not discussed the drawbacks of using the Bang-Bang controller. It was discussed earlier that using controllers are not usually ”free”, there is some cost of using them. If only the maximum magnitude of the controllers are used then this might be expensive to use, especially if the cost is quadratic. It will be up to the practitioner to decide whether the improved error is worth it.

Even though the Bang-Bang controller yields a better approximation of the dynamical systems, it might not be optimal when the agent then have to choose action in the environment. From *Figure 2* we can see that even though the agents choose action at the extremes, they are not strictly at the two extremes. So if we assume that that these algorithms are close to the optimal controller, it will not be a strictly Bang-bang controller. But still, using the Bang-bang controller during a the process of learning the dynamical system will yield a better approximation if we disregard the cost of using the Bang-Bang controller during the learning process.

References

[1] Johannes Traa. URL `https://www.scribd.com/document/551082766/matrix-calculus`. Matrix Calculus - Notes on the Derivative of a Trace.

[2] N. Matni and S. Tu. A tutorial on concentration bounds for system identification. *arXiv preprint*, 2019. URL `https://arxiv.org/abs/1906.11395`.

[3] N. Matni, A. Proutiere, and A. R. S. Tu. From self-tuning regulators to reinforcement learning and back again. *arXiv preprint*, 2019. URL `https://arxiv.org/abs/1906.11392`.

[4] Northeastern University, Course ME7247 Advanced Control Engineering. URL `https://laurentlessard.com/teaching/7247-advanced-control-engineering/`. Lecture notes by Laurent Lessard, Lecture 14.

[5] Private communications with Yishao Zhou (Thesis Supervisor). Explanations and guiding.

[6] T. Seyde, I. Gilitschenski, W. Schwarting, B. Stellato, M. Riedmiller, M. Wulfmeier, and D. Rus. Is bang-bang control all you need? solving continuous control with bernoulli policies. *arXiv preprint*, 2021. URL `https://arxiv.org/abs/2111.02552`.

[7] E. D. Sontag. *Mathematical Control Theory: Deterministic Finite-Dimensional Systems*. Springer-Verlag New York, Inc., 1998.

[8] Stockholm University, Course MM7024 - Linear Algebra and Learning from Data. Course material and lecture notes.

[9] Stockholm University, Course MM7027 - Dynamic Systems and Optimal Control Theory. Course material and lecture notes.

Thesis Misprints Corrections and Additional clarifications

Alexander Westberg

August 2023

Chapter 2 Proof of Proposition 2.3

The summation in

$$x^\top VDV^\top x = y^\top Dy = \sum_{i=1}^{n} \lambda_i \cdot y_i^2 > 0.$$

should be corrected to

$$\sum_{i=1}^{n} \lambda_{n-i} \cdot y_i^2.$$

since the eigenvalues are ordered in descending order starting from the top in D.

Chapter 2 Proof of Proposition 2.3

If a matrix has *strictly* positive eigenvalues, it is invertible. Assume there exists a vector such that $Ax = 0 = 0 \cdot x$, meaning that A is not invertible and zero would be an eigenvalue. Since A only has strictly positive eigenvalues, $Ax = 0 = 0 \cdot x$ is a a contradiction, since 0 is not positive.

Chapter 2 Proof of Proposition 2.7

When "Dimension" is mentioned during this proposition, it refers to the dimension of the columnspace of A, or simply *Rank* of A.

Chapter 2 Proof of Proposition 2.7

If A is invertible then $A^i(A^\top)^i$ is a positive definite matrix for any $i \geq 0$. Since

$$x^t A^i (A^\top)^i x = (A^i x)^\top (A^\top)^i x = ||(A^\top)^i x||_2^2 > 0.$$

If A is not invertible then there exist some x such that $(A^\top)^i x = (A^\top)^{i-1}(A^\top)x = 0$ which means that

$$x^t A^i (A^\top)^i x = ||(A^\top)^i x||_2^2 \geq 0,$$

meaning that $A^i(A^\top)^i$ is a semi positive definite.

Chapter 4 page 37

The function $c(u_t, x_t, x_{t+1})$ in

$$V^\pi = E\left[\sum_{t=0}^{T} \gamma^t c(u_t, x_t, x_{t+1}) + c_T(x_T)\right]$$

should corrected to $c(u_t, x_t)$.

Chapter 4 page 37

The summation

$$V^\pi = E\left[\sum_{t=0}^{T} \gamma^t c(u_t, x_t, x_{t+1}) + c_T(x_T)\right]$$

from 0 to T should be corrected to 0 to $T-1$.

Chapter 5.2.2 page 50

The inequality

$$\frac{1}{\lambda_1(\Sigma_{x_B}^{1/2})} < \frac{1}{\lambda_1(\Sigma_{x_G}^{1/2}) + 3\lambda_1(\Lambda_C(A,B,T))} \leq \frac{1}{\lambda_1(\Sigma_{x_G}^{1/2})}$$

is not correct.

What we want to show is that

$$\lambda_1(\Sigma_{X_B}^{-1/2}) < \lambda_1(\Sigma_{X_G}^{-1/2}),$$

where Σ_{X_B} and Σ_{X_G} is respectively the covariance matrix for the Bang-Bang controller and the Gaussian controller.

First we need to find the formula for $||M^{-1/2}||_2$ where M is positive definite matrix.

Since $||\cdot||_2$ is sub-multiplicative we have that

$$||M^{-1/2}||_2 = ||M^{-1}M^{1/2}||_2 \leq ||M^{-1}||_2||M^{1/2}||_2.$$

Now from the Linear Algebra preliminaries, we have that

$$||M^{-1}||_2 = \frac{1}{\sigma_1}$$

where $0 < \sigma_1 \leq ... \leq \sigma_n$ is the singular values of M.

So it remains to find $||M^{1/2}||_2$.
With the spectral theorem we have that $M = VDV^\top$ where where $V \in \mathbb{R}^{n\times n}$ is an orthonormal matrix where columns are eigenvectors of A and $D \in \mathbb{R}^{n\times n}$ is

a diagonal matrix, where the diagonal elements are the eigenvalues of A. Now $M^{0.5} = VD^{0.5}V^\top$ since

$$(VD^{0.5}V^\top)(VD^{0.5}V^\top) = VDV^\top.$$

So, it must be that

$$||M^{0.5}||_2 = \sqrt{\sigma_n}.$$

To conclude,

$$||M^{-1/2}||_2 \leq ||M^{-1}||_2||M^{0.5}||_2 = \frac{\sqrt{\sigma_n}}{\sigma_1}.$$

Under this subsection (5.2.2) we proved that the difference between $R(aM_1 + M_2, x)$ and $R(M_1 + M_2, x)$ is at minimum

$$R((a-1)M_1, x),$$

and also that

$$\lambda_1(M_1) + \lambda_1(M_2) \leq R(M_1 + M_2, x) \leq \lambda_n(M_1) + \lambda_n(M_2).$$

where $R(\cdot, x)$ is the Rayleigh quotient.

We denote $M_1 = \sigma_u^2 \Lambda_C(A, B, T)$ and $M_2 = \sigma_w^2 \Lambda_C(A, I, T)$. If we set the Gaussian controller to

$$\Sigma_{x_G} = \sigma_u^2 \Lambda_C(A, B, T) + \sigma_w^2 \Lambda_C(A, I, T)$$

then the Bang-Bang controller will be

$$\Sigma_{x_B} = (2 \cdot \sigma_u)^2 \Lambda_C(A, B, T) + \sigma_w^2 \Lambda_C(A, I, T) = 4 \cdot \sigma_u^2 \Lambda_C(A, B, T) + \sigma_w^2 \Lambda_C(A, I, T).$$

Now we have that that the minimum difference between

$$\lambda_1(\Sigma_{X_B}^{-1/2}) \quad \text{and} \quad \lambda_1(\Sigma_{X_G}^{-1/2})$$

must be

$$\frac{\sqrt{(a-1)\lambda_1(M_1) + \lambda_n(M_1 + M_2)}}{(a-1)\lambda_1(M_1) + \lambda_1(M_1 + M_2)} < \frac{\sqrt{\lambda_n(M_1 + M_2)}}{\lambda_1(M_1 + M_2)}$$

we input $a = 4$ and get

$$\frac{\sqrt{3\lambda_1(M_1) + \lambda_n(M_1 + M_2)}}{3\lambda_1(M_1) + \lambda_1(M_1 + M_2)} < \frac{\sqrt{\lambda_n(M_1 + M_2)}}{\lambda_1(M_1 + M_2)}.$$

Therefore the approximation error for the matrix A when using the Bang-Bang controller will be smaller compared to the Gaussian controller.