Optimal Control of Continuous and Discrete Time Systems via Generating Functions

Dissertation

Submitted in partial fulfilment of the requirements
for the degree of Doctor of Philosophy of Nagoya University

by

Dijian Chen

Nagoya University

July 2016

Abstract

Optimal control which deals with the problem of finding a control law for a given system such that a certain optimality criterion is achieved, is one of the dynamic optimization techniques popularly used in robotics, computer science, and operations research. There are two major tools for studying optimal control problems, one is the minimum principle, and the other the dynamic programming. After decades of the development, there have had many other methods for studying optimally controlled systems, among which the recently proposed generating function method which exhibits theoretical insights in solving optimal control problems and practical implication for real world applications attracts increasing attention of the researchers. In theory, this method thoroughly exploit optimal control problems' geometric structures, by utilizing Hamiltonian systems' characteristics, e.g. canonical transformation, symmetry, symplecticity, and so on. In practical computation, the method moves a large amount of computational effort to the off-line part such that it is substantially useful in on-line solutions repetitive generation for different state boundary conditions.

So far, the generating function method has been studied in existing literature to solve a small number of problems that there still has a large space for this thesis to develop the related theory and extend the method for solving other typical problems in continuous and discrete time cases, including extending the generating function method to solve continuous-time state constrained problems, developing the double generating functions method for discrete-time LQ optimal control with numerical stability analysis of the optimal generators, and solving the discrete-time nonlinear optimal control problems via generating functions.

First, this thesis extends the generating function approach to optimal control problems with path and terminal state constraints. We design a penalized problem by employing penalties that can converge to the original constrained problem under a mild condition, and prove that if such employed penalties satisfy a sufficient condition, the generating function coefficients can be solved recursively. Based on these two results, generating function method enables us to successfully solve the penalized problem instead of the constrained problem to obtain approximate solutions. Finally, we summarize how to design penalties suitable for the generating function method and gives the algorithm for different boundary conditions.

Second, this thesis develops the double generating function approach to discrete-time LQ optimal control problems. This method gives optimal generators only in terms of pre-computed coefficients and boundary conditions that is useful for the on-line repetitive computation for different boundary conditions. Moreover, since each generator contains inverse terms, the invertibility analysis is also performed to conclude that the terms in the generators constructed by double generating functions with opposite time directions are invertible under some mild conditions, while the terms with the same time directions will become singular when the time goes

infinity which may cause instabilities in numerical computations.

Last, this thesis develops the generating function approach to discrete-time nonlinear optimal control problems. This method gives optimal input analytically as state feedforward control in terms of the generating function. Since the generating function is nonlinear, we also develop numerical implementations to find its Taylor series expression in tensor notations. This finally gives optimal solutions expressed only in terms of the pre-computed generating function coefficients and state boundary conditions, such that it is useful for the on-demand optimal solutions generation for different boundary conditions.

Acknowledgement

There are a number of individuals and organizations without whom I would not have made it to the end of this thesis. I would like to express my sincere thanks to all of them.

Foremost, I would like to express my sincere gratitude to my two great advisors, Professor Kenji Fujimoto of Kyoto University and Professor Tatsuya Suzuki of Nagoya University. I would like to thank Professor Kenji Fujimoto for the continuous support of my Ph.D study and research, for his patience, motivation, enthusiasm, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis. I could not have imagined having a better advisor and mentor for my Ph.D study. I would like to thank Professor Tatsuya Suzuki for the successive support towards the completion of my studies in Nagoya, for his encouragement, insightful comments, and valuable suggestions. I am truly grateful to them.

I would like to express my sincere thanks to the senior alumni, Dr. Zhiwei Hao, now the Lecturer of Harbin Institute of Technology, for his concrete guidance in my early research and help in my daily life.

I would like to express my sincere gratitude to China Scholarship Council for its concrete sponsorship. I would like to thank Nagoya University, Kyoto University, and Nagoya International Center for providing me comfortable dormitories.

I would like to express my sincere thanks to Associate Professor Shinkichi Inagaki and Assistant Professor Hiroyuki Okuda of Nagoya University, especially the former faculty member Dr. Yuichi Tazaki now the Associate Professor of Kobe University, for their providing convenience in my research and laboratory common issues. I am thankful to all the faculty members and students of Suzuki laboratory of Nagoya University and Fujimoto laboratory of Kyoto University.

I would like to express my sincere gratitude to Professor Yoji Uno and Associate Professor Toru Asai of Nagoya University and Professor Noboru Sakamoto of Nanzan University for their careful review and thoughtful comments on my thesis.

Lastly, I am grateful to my fiancee Yan Zhao and to my parents Baixuan Chen and Yajuan Mao for their patient and warm encouragement.

Contents

Chapter 1

Introduction

In mathematics, computer science and operations research, mathematical optimization (alternatively, mathematical programming or simply, optimization) consists of maximization or minimization of a real function by systematically choosing input values within an allowed set and computing the value of the function. There are two categories, the static optimization and dynamic optimization. The static optimization makes choice at a single point of time. The textbook of S. Boyd [1] provides the fundamental and comprehensive results of this research field. The dynamic optimization deals with the problem over the time [2, 3, 4, 5]. Optimal control theory, a mathematical optimization method for deriving control policies, is a special case of the dynamic optimization. On the other hand, optimal control can also be seen as a control strategy in control theory. There are two major tools for studying optimal control problems. One is the minimum principle [6], formulated in 1956 by L.S. Pontryagin, is an extension of the variational principle. The other one is the dynamic programming [7] which was pioneered in the 1950s by R.E. Bellman. After decades of the development, there are many classical textbooks [8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20] for concluding the basic results of the theory, and also many new methods/tools [57, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32] for studying optimally controlled systems. Particularly, since the initial time of optimal control problems is fixed, then they can be classified into two kinds of problems according to the terminal time. The one whose terminal time is taken in the limit ∞ is called as the infinite horizon optimal control problem, while the one whose terminal time is also finitely fixed the finite horizon problem. The former infinite horizon problem is wildly used in industry applications [33]. In this thesis, we study the latter finite horizon problems which can be well solved by the model predictive control strategy [34]. However, since model predictive control is based on iteration, it increases the online computational burden.

According to Pontryagin's minimum principle, the necessary conditions for optimizing a dynamic system can be treated as a standard Hamiltonian system with state-costate variables (two point boundary value problem). The application of Hamiltonian mechanics can thoroughly exploit optimal control problems' geometric structures, by utilizing Hamiltonian systems' characteristics, e.g. canonical transformation, symmetry, symplecticity, and so on. Recently in 2004, V.M. Guibout [35] proposed a new method by using generating functions also in the framework of Hamiltonian mechanics to solve the two point boundary value problem for the spacecraft formation. Later, the generating function method was deeply studied by C. Park [36, 37] to exhibit

theoretical insights in solving optimal control problems and practical implication for aerospace applications. It is presented that the two point boundary value problem can be simply solved by algebraic manipulations of the generating functions, while exhibiting a method to solve optimal control problems with various kinds of boundary conditions. Based on this, Z. Hao [38, 39, 40] used a pair of different generating functions to generate optimal solutions. This method called as the double generating functions method dramatically reduces the on-line computational effort for different boundary conditions. Further for nonlinear problems, numerical implementations are required to obtain approximate generating functions. For this field of research see [41, 42, 43]. There also have generating function related researches in H_∞ control [44], receding-horizon control [45], optimal control problem with parameter variation [46], and so on [47]. These papers are all about the continuous-time case. For the discrete-time optimal control see [48, 49, 50]. Besides, application of the generating function method see [51, 52, 53, 54, 56].

We will investigate these papers deeply in the following five items.

- Continuous and discrete time problems

  There are more existing papers about the continuous-time case than those about the discrete-time case. All the papers above except [48, 49, 50] focus on solving the continuous-time problems, including LQ optimal control problem, nonlinear optimal control problem, numerical techniques of reducing Hamilton–Jacobi equation into ordinary differential equations for generating function coefficients, and so on. While in the discrete-time case, the paper [48] developed a discrete analogue of the Hamilton–Jacobi theory in mechanics that provides an appropriate way to study the discrete-time optimal control problem via generating fucntions. Based on this, the paper [50] developed the generating function method for the discrete-time LQ optimal control problem, and applies it to the partitioned hybrid systems. According to the investigation, there still has a large space for the development of generating function method in the research field of discrete-time problems.

- Unconstrained and constrained problems

  All the existing generating function related research papers above considered unconstrained problems, none of them took account of inequality path constraints. Other than direct methods, the generating function method belongs to indirect methods that are not good at handling inequality path constraints. The reason is that for such methods a priori knowledge of the optimal solution's structure is required, which is difficult to be attained. Since the research on constrained optimal control via generating functions is still blank, it is a research field of significant potentiality.

- Single and double generating function(s) methods

  As introduced, the single generating function method [36, 37, 50] uses only one generating function to give optimal input as state feedback control in terms of generating function with boundary conditions of the state. Since the generating function can be obtained off-line, this method performs efficient in on-line solutions generation for different boundary conditions by integrating the dynamics. In order to further reduce the on-line computational effort, the double generating functions method [38, 39, 40] is proposed that it uses a pair of different generating functions to give optimal solutions as algebraic expressions of generating functions with state boundary conditions. Due to this structure, the method

only needs to perform algebraic manipulations on-line without integrating the dynamics, such that it is more efficient in the on-line computation. Since such method is only developed for the continuous-time case, it is worth for researchers to develop the discrete analogue of the double generating functions method.

- Numerical stability analysis of optimal generators

  The optimal generators constructed by single/double generating function(s) method contains inverse terms for both continuous and discrete time cases. If the singularity would occur at some time steps or periods, it will cause the numerical instabilities. Therefore, the numerical stability analysis of optimal generators should be preformed to help us select the numerical stable generators. So far, only the paper [39] has given some preliminary analysis to show that the developed generators for optimal solutions constructed by double generating functions with the same time directions will cause instabilities when the time interval increases. This field of research can be of interest for the researchers.

- Numerical implementations of solving Hamilton–Jacobi equation

  For nonlinear problems, since the Hamilton–Jacobi equation is a nonlinear partial differential equation, it is difficult to find its analytic solution so that we need numerical implementations to find its approximate solution. So far, there have had two numerical implementations utilized for such a purpose. One is the Galerkin spectral technique with Chebyshev polynomials [41], and the other is the Taylor series expansion technique [35, 51, 43]. The first technique has the advantage of big region of convergence, but it also has the disadvantage that it requires the Hamiltonian for the optimal control problem has a special form and can not achieve the recursiveness for the generating function coefficients. The second technique has the advantage of recursive properties, but it also has the disadvantage that it is only applicable to systems that are close to linear systems, and it is inherently tied to the convergence of a power series for which it is difficult to estimate the region of convergence. There is a trade-off between these two techniques, so it is necessary for us to select the appropriate numerical implementation based on the comprehensive and deep evaluation of the problems.

Based on the above deep investigation, this thesis is interested in extending the generating function method to solve continuous-time state constrained problems, developing the double generating functions method for discrete-time LQ optimal control with numerical stability analysis of the optimal generators, and solving the discrete-time nonlinear optimal control problem via generating functions.

1.1 Continuous-time state constrained LQ optimal control

There exist two representative optimal control problem formulations, which are characterized by the types of terminal boundary conditions. One is called as the Hard Constraint Problem that in its problem setting the terminal boundary condition for state is pre-specified to a fixed point, while the other the Soft Constraint Problem that the terminal boundary condition for state is not pre-specified, but is indirectly affected by minimizing the final time performance index [36]. In this thesis, we focus on the former Hard Constraint Problem, which is more difficult to

solve than the latter Soft Constraint Problem but definitely common and significant in the engineering fields. Typical applications include the optimal rendezvous problem for the spacecraft [35, 51, 52, 53], optimal gait generation for the biped walking robot [55, 56], and so on. Many of conventional techniques work well to solve the Soft Constraint Problem, but are unavailable for the Hard Constraint Problem. The recent technique called the generating function method [36, 37] is developed particularly for the Hard Constraint Problem.

The generating function method is one of the indirect methods which solves a two point boundary value problem indirectly, instead of the original optimal control problem based on the minimum principle. In the two point boundary value problem, unlike Hamilton's equations describing the time evolution of the state-costate, the generating function specifies coordinate transformations of the state-costate from the boundary conditions. According to this, optimal input can be given as the state feedback control in terms of the pre-computed coefficients and the boundary conditions such that it is good at tackling the Hard Constraint Problems. In addition, since the calculation of the coefficients can be implemented off-line, this method reduces the on-line burden and is useful in the repetitive computation for a large numbers of different boundary conditions. This is another advantage of the generating function method. The generating function method was studied and applied to Hard Constraint Problem first time by the paper [36] that it proposed a framework for the optimal control by generating functions on the theoretical side. On the computational side, the key point of the method is to solve the Hamilton–Jacobi equation for the generating function numerically. Taylor series expansion is the most popular technique for this purpose. The papers [35, 36] first time used this technique to calculate the generating function approximately. Further, the paper [43] deeply investigated the Hamilton–Jacobi equation and presented in detail how to solve it successfully, which also made contributions. However, all these generating function based methods are for the Hard Constraint Problems without the general inequality path constraints. This limits the comprehensive application of the generating function method.

Other than direct methods, the generating function method belongs to indirect methods that are not good at handling inequality path constraints. The reason is that for such methods a priori knowledge of the optimal solution's structure is required [10], which is difficult to be attained. In order to extend the indirect methods to the inequality constrained problems, the natural idea is to convert the constrained problem to an unconstrained one such that we can avoid dealing with constraints. The penalty function (barrier function) which is well developed in the mathematical optimization [1] is a good candidate. There are many related theories and algorithms available for optimization problems. However, the application of the penalty technique to optimal control problems is few. The paper [57] applied an inverse penalty to the Mayer type optimal control problem with inequality constraints and shows the convergence of the minimum value under some mild conditions. The paper [58] extended this technique to the Lagrange type problem with inequality constraints and shows additional convergence of the state and input. The paper [59] extended the indirect shooting algorithm to the Bolza type problem with input inequality constraints by utilizing the logarithmic penalties, and proves the convergence for the LQ case and a specific nonlinear case. However, all these papers are for the Soft Constraint Problems that can not be readily extended to the generating function method for the Hard Constraint Problems.

1.2 Discrete-time LQ optimal control

After years of development, there exist many methods that work well to solve optimal control problems, e.g. dynamic programming [7], Riccati framework [60], and so on. However, most of these methods do not pay much attention to the computational effort when dealing with a large numbers of different boundary conditions which is common in the real applications. For example, the on-demand control of biped walking robot in the complex environment needs to adjust the step length and walking speed for each step [55, 56]. The conventional methods have to implement the whole computation repetitively for each different boundary conditions. This leads to heavy computational burdens.

Recently, the single generating function method for optimal control problems in continuous-time case has been proposed [36]. This method gives the optimal input as state feedback control with explicit pre-computed coefficients and boundary conditions. Such a structure enables us to calculate coefficients off-line in advance, and to generate optimal solutions by integrating the system equation on-line. From this viewpoint, it is useful for on-line repetitive computation of optimal solutions for different boundary conditions. In order to further reduce the on-line computational effort, the double generating function method was proposed [38, 39]. Compared with the single method, the double generating functions method gives the optimal solutions as algebraic equations in terms of pre-computed coefficients and boundary conditions based on a pair of different generating functions. Hence in the on-line computation, we only need to read saved coefficients and each set of boundary conditions to generate optimal solutions by algebraic manipulations without integrating the system equation. This method doubles the off-line work, but appears more efficient in on-line computation. Moreover, the paper [39] also gave some preliminary analysis to show that the developed generators for optimal solutions constructed by double generating functions with the same time directions will cause instabilities when the time interval increases.

Interesting characteristics of the generating function in continuous-time case also attract researchers to investigate the analogue in discrete-time case. In the field of mechanics, the paper [48] developed a discrete analogue of Hamilton–Jacobi theory that it provided an appropriate way to study the discrete-time optimal control problem via generating functions. The papers [49, 50] developed the single generating function method for the discrete-time LQ optimal control problem with an application to hybrid system. Though it is similar to the continuous-time case that the optimal input is also given as state feedback control, but this discrete analogue [49, 50] has to compute coefficients of two different generating functions off-line and will cause instabilities since the term needs to be inverted in the developed generator is singular. These two problems limits the application of the discrete single generating function method.

1.3 Discrete-time nonlinear optimal control

Optimal control deals with the problem of finding a control law for a given system such that a certain optimality criterion is achieved. After years of development, there exist many classical methods to solve optimal control problems, e.g. the dynamic programming [7], the shooting algorithm [61], and so on. However, most of these methods do not pay much attention to the computational effort when dealing with a number of different boundary conditions which

is common in the real applications. For example, the optimal spacecraft rendezvous with different initial positions [52]. The traditional methods have to implement the whole computation repetitively for each different boundary position. This leads to heavy computational burdens.

Recently, the generating function method has been proposed for the continuous-time optimal control problems [36, 37]. This method can provide the optimal input analytically as state feedback control in terms of the boundary conditions. Due to this structure, it is useful for us to efficiently generate optimal solutions for different boundary conditions. This is the advantage of the generating function method in contrast with conventional methods. Further, for the analytical solutions given by the generating function method, the key point is to find the corresponding generating function which satisfies the Hamilton–Jacobi equation. It is verified for the LQ case that the related generating function also takes the quadratic form [35]. However for the general nonlinear cases, it is almost impossible to find the explicit expression of generating function by solving the nonlinear partial differential Hamilton–Jacobi equation. The numerical implementations are needed to find its approximate expression. The paper [41] employed the Galerkin spectral technique with Chebyshev polynomials to solve the Hamilton–Jacobi equation for the generating functions. However, this technique requires that the Hamiltonian for the optimal control problem has a special form and does not possess the recursiveness. The paper [43] gave an algorithm via Taylor series expansion with Kronecker product to reduce the nonlinear partial differential Hamilton–Jacobi equation to ordinary differential equations that allowed one to solve them recursively for the generating function coefficients. However, this framework using Kronecker product notation can not well handle the Hamilton–Jacobi equation such that it can not reduce the Hamilton–Jacobi equation into difference equations for generating function coefficients.

Interesting characteristics of the generating function method for the continuous-time problems also attract researchers to investigate the analogue for the discrete-time cases. The paper [48] developed a discrete analogue of Hamilton–Jacobi theory in the mechanics field provides an appropriate way to study the discrete-time optimal control problem via generating functions. Based on this, The papers [49, 50] developed the generating function method for the discrete-time LQ optimal control problem, and applied it to the partitioned hybrid systems. For the research on discrete-time nonlinear optimal control via generating functions, it is still blank.

1.4 Goals and contributions of the thesis

This thesis has three main contributions in extending the generating function method to solve continuous-time state constrained problems, developing the double generating functions method for discrete-time LQ optimal control with numerical stability analysis of the optimal generators, and solving the discrete-time nonlinear optimal control problem via generating functions. They are stated in detail in the following.

Firstly, this thesis extends the generating function method to the Hard Constraint Problem with inequality path constraints, i.e. the classical path and terminal state constrained optimal control problem [10]. First, we formulate and design the constrained problem and the penalized Hard Constraint Problem respectively, show their convex properties, and further exhibit the convergence of the minimum cost function value and optimal solutions between these two problems under a mild condition. Second, due to the technique of Taylor series expansion, the partial

differential Hamilton–Jacobi equation is reduced to ordinary differential equations for the generating function coefficients. We give the recursive condition to eliminate the coupling relations between the coefficients with lower and higher indices in these ordinary differential equations so that they can be solved recursively. This guarantees the penalized Hard Constraint Problem can be successfully solved by the generating function method. Based on this, we summarize how to design penalties which is suitable for the generating function method, and gives an algorithm presents how to generate optimal solutions repetitively for different boundary conditions.

Secondly, this thesis develops the discrete analogue of double generating function method. To clearly present the fundamental feature of this method and to make it convenient for further extension to the nonlinear problems, this thesis investigates the classical discrete-time LQ optimal control problem. First, we derive the left discrete Hamiltonian, Hamilton's equations, and the Hamilton–Jacobi equation for the LQ optimal control problem which is a counterpart to the right ones in the references [49, 50] according to discrete mechanics [48]. Second, we choose appropriate Hamilton–Jacobi equation, left or right, to solve for the forward type II, III, and backward type III generating functions[†]. Then by selecting any two different generating functions from the four single ones, we can construct six double generating functions which give six generators for optimal solutions, respectively. These discrete generators maintain the advantage of on-line efficient computation for different boundary conditions, which is presented by a followed algorithm. Besides, since each generator contains inverse terms, we deeply perform the numerical stability analysis to conclude that the terms in the generators constructed by double generating functions with opposite time directions are invertible under some mild conditions, while the terms with the same time directions will become singular when the time goes infinity which may cause instability in numerical computations.

Thirdly, this thesis develops the generating function method for the general discrete-time nonlinear optimal control problems. First, we give the analytically optimal solutions, which is expressed as the state feedforward control in terms of the generating functions. Then in the numerical implementations, we systematically perform three steps to solve the Hamilton–Jacobi equation for the generating functions. In detail, we expand all the nonlinear functions in the Hamilton–Jacobi equation as Taylor series about zeros in tensor notations such that they can clearly present the detailed structure of the Hamilton–Jacobi equation later during the reduction. Based on this, we again employ the Taylor series technique to successfully replace one variable by the other two in the Hamilton–Jacobi equation to rewrite it by the addressed theorem in the thesis. Due to this step, we achieve our objective that the Hamilton–Jacobi equation is reduced to the difference equations for the generating function coefficients, and they can be solved recursively with respect to the order of the Taylor series. The developed numerical framework can give the optimal solutions in terms of the pre-computed generating function coefficients and boundary conditions, such that we can divide the whole computation into two parts, the off-line part calculates the coefficients in advance, and the on-line part efficiently generates optimal solutions for different boundary conditions. From this viewpoint, it is useful for the on-demand optimal solutions generation for different boundary conditions.

[†]Basically, there exist four types of generating functions, type I, II, III, and IV, and each type also has two kinds, forward and backward [36, 39].

1.5 Organization of the thesis

Chapter 2 introduces preliminaries of the Hamiltonian system and the generating functions. For the continuous and discrete time optimal control problems, we give necessary and sufficient (only for continuous-time case) conditions for optimality, derive Hamilton–Jacobi equations and generating functions, provide optimal solutions (only for continuous-time case), present relations between the generating function and the value function, and exhibit the LQ cases.

Chapter 3 studies the continuous-time state constrained optimal control problem via generating functions. It first formulates the original and penalized problems and exhibits their convex and convergent properties, then introduces the generating function approach, shows the recursive condition that enables us numerically solve the Hamilton–Jacobi equation, and gives the design principle of the penalty and the algorithm for different boundary conditions. At last, two examples are presented to illustrate the effectiveness of the developed method.

Chapter 4 studies the discrete-time LQ optimal control problem via double generating functions. It first formulates the discrete-time LQ optimal control problem and introduces the necessary conditions for optimality. Based on this, it derives the forward and backward generating functions, and develops generators for optimal solutions. Furthermore, it also performs the numerical stability analysis. At last, two examples are presented to illustrate the effectiveness of the developed method.

Chapter 5 studies the discrete-time nonlinear optimal control problem via generating functions. It first formulates the discrete-time nonlinear optimal control problem and derives the analytically optimal solutions via the generating function. Then, it addresses the Taylor series based numerical implementations to give numerical generating functions and optimal solutions. At last, two examples are presented to illustrate the effectiveness of the developed method.

Chapter 6 presents a brief conclusion of the research carried out in this thesis. This is followed by summarizing remarks and suggestions for the future research.

Chapter 2

Hamiltonian system and generating functions

According to Pontryagin's minimum principle, the necessary conditions for optimizing a dynamic system can be considered as a standard Hamiltonian system for the state-costate variables. This method thoroughly exploits optimal control problems' geometric structures, by utilizing Hamiltonian systems' characteristics, e.g. canonical transformation, symmetry, symplecticity, and so on [70].

In Hamiltonian system, the generating function satisfying Hamilton–Jacobi equation specifies a family of canonical transformations from boundary state-costate to current state-costate that describe the dynamics of state-costate defined by Hamilton's equations [48]. This recent developed generating function framework [35, 51, 36, 52, 37, 41, 39, 48, 50] exhibits theoretical insights and practical implication in solving continuous and discrete time optimal control problems by using generating functions.

In order to enrich and develop the generating function method, we introduce the preliminaries of Hamiltonian system and generating functions in this chapter. Particularly, Section 2.1 introduces the continuous-time case, where we formulate the continuous-time optimal control problem, give the necessary and sufficient conditions for optimality in Section 2.1.1 by referring to the lecture note of B. Chachuat [62]. Based on this, in Section 2.1.2, we derive the Hamilton–Jacobi equation and generating function via coordinate transformations in the Hamiltonian system by referring to H. Goldstein's classical textbook [63]. In Section 2.1.3, we give optimal solutions via both single generating function method by C. Park [36] and double generating functions method by Z. Hao [40]. Though the generating function is derived under Pontryagin's minimum principle, it should has relations with the value function which is central to the dynamic programming that is another major tool for studying optimally controlled systems. This is introduced in Section 2.1.4 by referring to the work of C. Park [36]. Finally in Section 2.1.5, we introduce the LQ case which can clearly exhibit the feature and advantage of the generating function method by referring to the work of Z. Hao [39].

Section 2.2 introduces the discrete-time case. Compared with the continuous-time case, there is fewer literature that concentrates on the discrete-time field though it possesses unique theoretical significance. First in Section 2.2.1, we give the necessary conditions for optimizing the formulated discrete-time optimal control problem by referring to the work of T. Ohsawa

[48]. Similarly in Section 2.2.2, we derive the discrete Hamilton–Jacobi equation and generating function by referring to the work of T. Lee [50]. For the relation with the value function, we ourself prove the related theorem in 2.2.3 to make the discrete-time part complete. Finally in Section 2.2.4, we also introduce the discrete-time LQ case by referring to the work of T. Lee [50].

2.1 Continuous-time case

Consider the following continuous-time optimal control problem.

Problem 2.1.

$$\min_{u} \int_{t_0}^{t_\mathrm{f}} \left(Q\left(x(t)\right) + \frac{1}{2} u(t)^\mathsf{T} R\left(x(t)\right) u(t) \right) \mathrm{d}t \tag{2.1}$$

$$\text{s.t. } \dot{x}(t) = A\left(x(t)\right) + B\left(x(t)\right) u(t), \quad t \in [t_0, t_\mathrm{f}] \tag{2.2}$$

$$x(t_0) = x_\mathrm{init}, \quad x(t_\mathrm{f}) = x_\mathrm{term} \tag{2.3}$$

where "s.t." is the abbreviation of the phrase "subject to", $x \in \mathbb{R}^n$ and $u \in \mathbb{R}^m$ are the state and input variables, respectively. The functions $Q\colon \mathbb{R}^n \to \mathbb{R}$, $R\colon \mathbb{R}^n \to \mathbb{R}^m \times \mathbb{R}^m$, $A\colon \mathbb{R}^n \to \mathbb{R}^n$, and $B\colon \mathbb{R}^n \to \mathbb{R}^n \times \mathbb{R}^m$. Moreover, the function $Q \succcurlyeq 0$[†], and the matrix $R(x) \succ 0, \forall x \in \mathbb{R}^n$. In addition, all the functions Q, R, A, and B are continuous in x and have continuous first partial derivatives with respect to x, $\forall x \in \mathbb{R}^n$. In (2.3), $x_\mathrm{init} \in \mathbb{R}^n$ and $x_\mathrm{term} \in \mathbb{R}^n$ are the given initial and terminal state values, respectively.

Problem 2.1 is a Hard Constraint Problem[‡], and the associated pre-Hamiltonian $\bar{H}\colon \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ is given by adjoining the right hand side of the differential equation in (2.2) to the cost integrand in (2.1) as

$$\bar{H}(x, \lambda, u) \colon = \left(Q(x) + \frac{1}{2} u^\mathsf{T} R(x) u \right) + \lambda^\mathsf{T} \left(A(x) + B(x) u \right) \tag{2.4}$$

where $\lambda \in \mathbb{R}^n$ is introduced as the costate. We denote the optimal state, costate, and input of Problem 2.1 as x^*, λ^*, and u^*, respectively.

For Problem 2.1, we make the following assumption.

Assumption 2.1. *Assume that in Problem 2.1, both the functions Q and R are (strictly) jointly convex in x, $\forall x \in \mathbb{R}^n$. Moreover, either the condition "A and B are (strictly) jointly convex in x, $\forall x \in \mathbb{R}^n$, and $\lambda^*(t) \geqslant 0$, $\forall t \in [t_0, t_\mathrm{f}]$" or the condition "$A$ and B are (strictly) jointly concave in x, $\forall x \in \mathbb{R}^n$, and $\lambda^*(t) \leqslant 0$, $\forall t \in [t_0, t_\mathrm{f}]$" holds.*

[†]The positive semi-definiteness of the function Q implies that $Q(0) = 0$ and $Q(x) \geqslant 0$ for every non-zero $x \in \mathbb{R}^n$.

[‡]Other than the conventional Soft Constraint Problem where $x(t_\mathrm{f})$ is not prescribed but indirectly affected by minimizing the terminal cost, Problem 2.1 is called as the Hard Constraint Problem where the terminal state is prescribed.

2.1.1 Necessary and sufficient conditions for optimality

The first-order necessary conditions for minimizing Problem 2.1 can be derived by the minimum principle which was formulated by the Russian mathematician L.S. Pontryagin in 1956. This is presented in the following theorem.

Theorem 2.1 ([62]). *For Problem 2.1, there is a vector function λ^* of class $\mathcal{C}^1$ such that the triple (x^*, λ^*, u^*), where x^* is of class $\mathcal{C}^1$ and u^* is of class $\mathcal{C}^0$, satisfies ($t \in [t_0, t_f]$)*

$$\dot{x} = \frac{\partial \bar{H}(x, \lambda, u)}{\partial \lambda}, \qquad x(t_0) = x_{\text{init}}, \quad x(t_f) = x_{\text{term}} \tag{2.5}$$

$$\dot{\lambda} = -\frac{\partial \bar{H}(x, \lambda, u)}{\partial x} \tag{2.6}$$

$$u = \arg\min_{\bar{u}} \bar{H}(x, \lambda, \bar{u}) \equiv -R(x)^{-1} B(x)^{\mathsf{T}} \lambda. \tag{2.7}$$

Note that the optimal input in (2.7) given by the necessary conditions is the local minimizer of Problem 2.1. Substitution of (2.7) into the pre-Hamiltonian (2.4) and the Hamilton's equations (2.5)–(2.6) gives the Hamiltonian system for the state and costate

$$H(x, \lambda) = Q(x) + \lambda^{\mathsf{T}} A(x) - \frac{1}{2} \lambda^{\mathsf{T}} B(x) R(x)^{-1} B(x)^{\mathsf{T}} \lambda \tag{2.8}$$

$$\dot{x} = \frac{\partial H(x, \lambda)}{\partial \lambda} \tag{2.9}$$

$$\dot{\lambda} = -\frac{\partial H(x, \lambda)}{\partial x}. \tag{2.10}$$

Further, we expect to determine the global minimizer that achieves the global minimum cost function value, not only the local minimizer that gives the local minima. Conditions under which the necessary conditions are also sufficient for minimizing Problem 2.1 is presented in the following theorem (called as Mangasarian sufficient conditions).

Theorem 2.2 ([62]). *Under Assumption 2.1, for Problem 2.1, if there is a vector function λ^* of class $\mathcal{C}^1$ such that the triple (x^*, λ^*, u^*), where x^* is of class $\mathcal{C}^1$ and u^* is of class $\mathcal{C}^0$, satisfies (2.5)–(2.7), then u^* is a (strict) global minimizer of Problem 2.1.*

As is presented, Theorem 2.2 requires the convexity or concavity of the functions in Problem 2.1 and also the sign of the optimal costate (Assumption 2.1).

2.1.2 Hamilton–Jacobi equation and generating function

In the Hamiltonian system (2.8)–(2.10), the state x and the costate λ are the canonical coordinates [63]. Now, consider the new canonical coordinates $\hat{x} \in \mathbb{R}^n$ and $\hat{\lambda} \in \mathbb{R}^n$ governed by the new Hamiltonian $\hat{H} \colon \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$. For the old and new Hamiltonians, there exists the following relation to connect them

$$\lambda^{\mathsf{T}} \dot{x} - H(x, \lambda) = \hat{\lambda}^{\mathsf{T}} \dot{\hat{x}} - \hat{H}(\hat{x}, \hat{\lambda}) + \frac{\mathrm{d}F}{\mathrm{d}t} \tag{2.11}$$

where $F\colon \mathbb{R}^n \times \mathbb{R}^n \times [t_0, t_f] \to \mathbb{R}$ is the generating function. By selecting one variable from the old coordinates and the other the new coordinates, there mainly have four kinds of generating functions $F_1(x, \hat{x}, t)$, $F_2(x, \hat{\lambda}, t)$, $F_3(\lambda, \hat{x}, t)$, and $F_4(\lambda, \hat{\lambda}, t)$. Here, we are only interested in the constant new coordinates, e.g. $(x(t_0), \lambda(t_0))$ and $(x(t_\mathrm{f}), \lambda(t_\mathrm{f}))$, that lead to zero Hamiltonian $\hat{H} = 0$. We call those with initial coordinate variables as forward generating functions (with "f" in the subscript), while terminal coordinate variables backward generating functions (with "b" in the subscript). In view of this, there totally have eight different kinds of generating functions $F_{1\mathrm{f}}(x, x(t_0), t)$, $F_{2\mathrm{f}}(x, \lambda(t_0), t)$, $F_{3\mathrm{f}}(\lambda, x(t_0), t)$, $F_{4\mathrm{f}}(\lambda, \lambda(t_0), t)$, $F_{1\mathrm{b}}(x, x(t_\mathrm{f}), t)$, $F_{2\mathrm{b}}(x, \lambda(t_\mathrm{f}), t)$, $F_{3\mathrm{b}}(\lambda, x(t_\mathrm{f}), t)$, and $F_{4\mathrm{b}}(\lambda, \lambda(t_\mathrm{f}), t)$. There exist Legendre transformations [36] between forward generating functions

$$F_{1\mathrm{f}}(x, x(t_0), t) = F_{2\mathrm{f}}(x, \lambda(t_0), t) - \lambda(t_0)^{\mathsf{T}} x(t_0) \tag{2.12}$$

$$F_{1\mathrm{f}}(x, x(t_0), t) = F_{3\mathrm{f}}(\lambda, x(t_0), t) + \lambda^{\mathsf{T}} x \tag{2.13}$$

$$F_{1\mathrm{f}}(x, x(t_0), t) = F_{4\mathrm{f}}(\lambda, \lambda(t_0), t) + \lambda^{\mathsf{T}} x - \lambda(t_0)^{\mathsf{T}} x(t_0) \tag{2.14}$$

and between backward generating functions

$$F_{1\mathrm{b}}(x, x(t_\mathrm{f}), t) = F_{2\mathrm{b}}(x, \lambda(t_\mathrm{f}), t) - \lambda(t_\mathrm{f})^{\mathsf{T}} x(t_\mathrm{f}) \tag{2.15}$$

$$F_{1\mathrm{b}}(x, x(t_\mathrm{f}), t) = F_{3\mathrm{b}}(\lambda, x(t_\mathrm{f}), t) + \lambda^{\mathsf{T}} x \tag{2.16}$$

$$F_{1\mathrm{b}}(x, x(t_\mathrm{f}), t) = F_{4\mathrm{b}}(\lambda, \lambda(t_\mathrm{f}), t) + \lambda^{\mathsf{T}} x - \lambda(t_\mathrm{f})^{\mathsf{T}} x(t_\mathrm{f}). \tag{2.17}$$

By substituting generating function into (2.11), we can get basic relations and Hamilton–Jacobi equations for the eight kinds of generating functions. This is presented in the following proposition.

Proposition 2.1 ([36, 40]). *For Problem 2.1*

(i) *The generating function $F_{1\mathrm{f}}(x, x(t_0), t)$ satisfying the Hamilton–Jacobi equation*

$$\frac{\partial F_{1\mathrm{f}}(x, x(t_0), t)}{\partial t} + H\left(x, \frac{\partial F_{1\mathrm{f}}(x, x(t_0), t)}{\partial x}\right) = 0 \tag{2.18}$$

specifies the family of forward canonical transformations $(x(t_0), \lambda(t_0)) \mapsto (x(t), \lambda(t))$, $t \in [t_0, t_\mathrm{f}]$, by the basic relations

$$\lambda = \frac{\partial F_{1\mathrm{f}}(x, x(t_0), t)}{\partial x} \tag{2.19}$$

$$\lambda(t_0) = -\frac{\partial F_{1\mathrm{f}}(x, x(t_0), t)}{\partial x(t_0)}. \tag{2.20}$$

(ii) *The generating function $F_{2\mathrm{f}}(x, \lambda(t_0), t)$ satisfying the Hamilton–Jacobi equation*

$$\frac{\partial F_{2\mathrm{f}}(x, \lambda(t_0), t)}{\partial t} + H\left(x, \frac{\partial F_{2\mathrm{f}}(x, \lambda(t_0), t)}{\partial x}\right) = 0 \tag{2.21}$$

specifies the family of forward canonical transformations $(x(t_0), \lambda(t_0)) \mapsto (x(t), \lambda(t))$, $t \in [t_0, t_\mathrm{f}]$, by the basic relations

$$\lambda = \frac{\partial F_{2\mathrm{f}}(x, \lambda(t_0), t)}{\partial x} \tag{2.22}$$

$$x(t_0) = \frac{\partial F_{2\mathrm{f}}(x, \lambda(t_0), t)}{\partial \lambda(t_0)}. \tag{2.23}$$

(iii) *The generating function* $F_{3\mathrm{f}}(\lambda, x(t_0), t)$ *satisfying the Hamilton–Jacobi equation*

$$\frac{\partial F_{3\mathrm{f}}(\lambda, x(t_0), t)}{\partial t} + H\left(-\frac{\partial F_{3\mathrm{f}}(\lambda, x(t_0), t)}{\partial \lambda}, \lambda\right) = 0 \tag{2.24}$$

specifies the family of forward canonical transformations $(x(t_0), \lambda(t_0)) \mapsto (x(t), \lambda(t))$, $t \in [t_0, t_\mathrm{f}]$, *by the basic relations*

$$x = -\frac{\partial F_{3\mathrm{f}}(\lambda, x(t_0), t)}{\partial \lambda} \tag{2.25}$$

$$\lambda(t_0) = -\frac{\partial F_{3\mathrm{f}}(\lambda, x(t_0), t)}{\partial x(t_0)}. \tag{2.26}$$

(iv) *The generating function* $F_{4\mathrm{f}}(\lambda, \lambda(t_0), t)$ *satisfying the Hamilton–Jacobi equation*

$$\frac{\partial F_{4\mathrm{f}}(\lambda, \lambda(t_0), t)}{\partial t} + H\left(-\frac{\partial F_{4\mathrm{f}}(\lambda, \lambda(t_0), t)}{\partial \lambda}, \lambda\right) = 0 \tag{2.27}$$

specifies the family of forward canonical transformations $(x(t_0), \lambda(t_0)) \mapsto (x(t), \lambda(t))$, $t \in [t_0, t_\mathrm{f}]$, *by the basic relations*

$$x = -\frac{\partial F_{4\mathrm{f}}(\lambda, \lambda(t_0), t)}{\partial \lambda} \tag{2.28}$$

$$x(t_0) = \frac{\partial F_{4\mathrm{f}}(\lambda, \lambda(t_0), t)}{\partial \lambda(t_0)}. \tag{2.29}$$

(v) *The generating function* $F_{1\mathrm{b}}(x, x(t_\mathrm{f}), t)$ *satisfying the Hamilton–Jacobi equation*

$$\frac{\partial F_{1\mathrm{b}}(x, x(t_\mathrm{f}), t)}{\partial t} + H\left(x, \frac{\partial F_{1\mathrm{b}}(x, x(t_\mathrm{f}), t)}{\partial x}\right) = 0 \tag{2.30}$$

specifies the family of backward canonical transformations $(x(t_\mathrm{f}), \lambda(t_\mathrm{f})) \mapsto (x(t), \lambda(t))$, $t \in [t_0, t_\mathrm{f}]$, *by the basic relations*

$$\lambda = \frac{\partial F_{1\mathrm{b}}(x, x(t_\mathrm{f}), t)}{\partial x} \tag{2.31}$$

$$\lambda(t_\mathrm{f}) = -\frac{\partial F_{1\mathrm{b}}(x, x(t_\mathrm{f}), t)}{\partial x(t_\mathrm{f})}. \tag{2.32}$$

(vi) *The generating function* $F_{2\mathrm{b}}(x, \lambda(t_\mathrm{f}), t)$ *satisfying the Hamilton–Jacobi equation*

$$\frac{\partial F_{2\mathrm{b}}(x, \lambda(t_\mathrm{f}), t)}{\partial t} + H\left(x, \frac{\partial F_{2\mathrm{b}}(x, \lambda(t_\mathrm{f}), t)}{\partial x}\right) = 0 \tag{2.33}$$

specifies the family of backward canonical transformations $(x(t_\mathrm{f}), \lambda(t_\mathrm{f})) \mapsto (x(t), \lambda(t))$, $t \in [t_0, t_\mathrm{f}]$, *by the basic relations*

$$\lambda = \frac{\partial F_{2\mathrm{b}}(x, \lambda(t_\mathrm{f}), t)}{\partial x} \tag{2.34}$$

$$x(t_\mathrm{f}) = \frac{\partial F_{2\mathrm{b}}(x, \lambda(t_\mathrm{f}), t)}{\partial \lambda(t_\mathrm{f})}. \tag{2.35}$$

(vii) *The generating function* $F_{3\mathrm{b}}(\lambda, x(t_\mathrm{f}), t)$ *satisfying the Hamilton–Jacobi equation*

$$\frac{\partial F_{3\mathrm{b}}(\lambda, x(t_\mathrm{f}), t)}{\partial t} + H\left(-\frac{\partial F_{3\mathrm{b}}(\lambda, x(t_\mathrm{f}), t)}{\partial \lambda}, \lambda\right) = 0 \tag{2.36}$$

specifies the family of backward canonical transformations $(x(t_\mathrm{f}), \lambda(t_\mathrm{f})) \mapsto (x(t), \lambda(t))$, $t \in [t_0, t_\mathrm{f}]$, *by the basic relations*

$$x = -\frac{\partial F_{3\mathrm{b}}(\lambda, x(t_\mathrm{f}), t)}{\partial \lambda} \tag{2.37}$$

$$\lambda(t_\mathrm{f}) = -\frac{\partial F_{3\mathrm{b}}(\lambda, x(t_\mathrm{f}), t)}{\partial x(t_\mathrm{f})}. \tag{2.38}$$

(viii) *The generating function* $F_{4\mathrm{b}}(\lambda, \lambda(t_\mathrm{f}), t)$ *satisfying the Hamilton–Jacobi equation*

$$\frac{\partial F_{4\mathrm{b}}(\lambda, \lambda(t_\mathrm{f}), t)}{\partial t} + H\left(-\frac{\partial F_{4\mathrm{b}}(\lambda, \lambda(t_\mathrm{f}), t)}{\partial \lambda}, \lambda\right) = 0 \tag{2.39}$$

specifies the family of backward canonical transformations $(x(t_\mathrm{f}), \lambda(t_\mathrm{f})) \mapsto (x(t), \lambda(t))$, $t \in [t_0, t_\mathrm{f}]$, *by the basic relations*

$$x = -\frac{\partial F_{4\mathrm{b}}(\lambda, \lambda(t_\mathrm{f}), t)}{\partial \lambda} \tag{2.40}$$

$$x(t_\mathrm{f}) = \frac{\partial F_{4\mathrm{b}}(\lambda, \lambda(t_\mathrm{f}), t)}{\partial \lambda(t_\mathrm{f})}. \tag{2.41}$$

Remark 2.1. The forward canonical transformation $(x(t_0), \lambda(t_0)) \mapsto (x(t), \lambda(t))$ at initial time $t = t_0$ is the identity transformation $(x(t_0), \lambda(t_0)) \mapsto (x(t_0), \lambda(t_0))$. It is clear from Proposition 2.1 that such identity transformation can be specified by the relations (2.22)–(2.23) of $F_{2\mathrm{f}}$, or the relations (2.25)–(2.26) of $F_{3\mathrm{f}}$. However, it can neither be specified by the relations (2.19)–(2.20) of $F_{1\mathrm{f}}$, nor be specified by the relations (2.28)–(2.29) of $F_{4\mathrm{f}}$. This implies the functions $F_{2\mathrm{f}}$ and $F_{3\mathrm{f}}$ are well-defined at initial time, while $F_{1\mathrm{f}}$ and $F_{4\mathrm{f}}$ are not well-defined at initial time. Similar for the backward generating functions, $F_{2\mathrm{b}}$ and $F_{3\mathrm{b}}$ are well-defined at terminal time, while $F_{1\mathrm{b}}$ and $F_{4\mathrm{b}}$ are not well-defined at terminal time. In summary, we have $F_{2\mathrm{f}}(x, \lambda(t_0), t)|_{t=t_0} = \lambda(t_0)^\mathsf{T} x(t_0)$, $F_{3\mathrm{f}}(\lambda, x(t_0), t)|_{t=t_0} = -\lambda(t_0)^\mathsf{T} x(t_0)$, $F_{2\mathrm{b}}(x, \lambda(t_\mathrm{f}), t)|_{t=t_\mathrm{f}} = \lambda(t_\mathrm{f})^\mathsf{T} x(t_\mathrm{f})$, and $F_{3\mathrm{b}}(\lambda, x(t_\mathrm{f}), t)|_{t=t_\mathrm{f}} = -\lambda(t_\mathrm{f})^\mathsf{T} x(t_\mathrm{f})$.

2.1.3 Optimal solutions via generating functions

As stated in Remark 2.1, the four generating functions $F_{2\mathrm{f}}$, $F_{3\mathrm{f}}$, $F_{2\mathrm{b}}$, and $F_{3\mathrm{b}}$ are well-defined[†] such that each of them can be employed to generate optimal solutions of Problem 2.1. Here, we only set the example of using $F_{2\mathrm{b}}$ (the others are similar), which is presented in the following theorem. Since only one generating function is used, we call the method as single generating function method.

Theorem 2.3 ([36]). *The optimal input of Problem 2.1 is given as the state feedback control*

$$u^*(t) = -R(x)^{-1}B(x)^{\mathsf{T}}\frac{\partial F_{2\mathrm{b}}(x,\lambda(t_{\mathrm{f}}),t)}{\partial x}, \quad t \in [t_0, t_{\mathrm{f}}] \tag{2.42}$$

where the terminal costate $\lambda(t_{\mathrm{f}})$ is determined by solving the following equation

$$x(t_{\mathrm{f}}) = \left.\frac{\partial F_{2\mathrm{b}}(x,\lambda(t_{\mathrm{f}}),t)}{\partial \lambda(t_{\mathrm{f}})}\right|_{t=t_0}. \tag{2.43}$$

In Theorem 2.3, if we can find the explicit expression of the generating function, we readily get the optimal input by (2.42). Since the Hamilton–Jacobi equation (2.33) is a nonlinear partial differential equation, it is difficult to get its analytic solution, i.e. the analytic generating function, so that we need the numerical implementations to get its approximate solution, for example the Galerkin spectral technique, the Taylor series expansion technique, and so on. For the details see [41, 43].

Further, a method using a pair of different generating functions (double generating functions method) is proposed in [40] to generate optimal solutions. This is presented in the following theorem (using the pair of $F_{3\mathrm{f}}$ and $F_{3\mathrm{b}}$).

Theorem 2.4 ([40]). *The optimal state and input of Problem 2.1 are given as*

$$x^* = -\left.\frac{\partial F_{3\mathrm{f}}(\lambda,x(t_0),t)}{\partial \lambda}\right|_{\lambda=\lambda^*} \text{ or } -\left.\frac{\partial F_{3\mathrm{b}}(\lambda,x(t_{\mathrm{f}}),t)}{\partial \lambda}\right|_{\lambda=\lambda^*} \tag{2.44}$$

$$u^* = -R(x^*)^{-1}B(x^*)^{\mathsf{T}}\lambda^* \tag{2.45}$$

respectively, where λ^* is the solution of the following equation

$$\frac{\partial}{\partial x}\left(F_{3\mathrm{f}}(\lambda,x(t_0),t) - F_{3\mathrm{b}}(\lambda,x(t_{\mathrm{f}}),t)\right) = 0. \tag{2.46}$$

Unlike the single generating function method presented in Theorem 2.3, the double generating functions method in Theorem 2.4 gives optimal input in terms of generating functions with state boundary conditions algebraically. This is useful for numerical computations. For the details see [40].

[†]The first and fourth kinds of generating functions can also be used to generate optimal solutions, prior to which they have to be obtained via Legendre transformations [36] from other well-defined generating functions as introduced in Chapter 2. This is relatively complicated, hence for the sake of convenience, we here only consider the well-defined generating functions.

2.1.4 Relation between generating function and value function

There are two major tools for studying optimally controlled systems. Besides the Pontryagin's minimum principle introduced in Section 2.1.1, the other one is the dynamic programming which was pioneered in the 1950s by R.E. Bellman.

The Hamilton–Jacobi–Bellman equation

$$\frac{\partial V(x,t)}{\partial t} = -\min_{u}\left(\left(Q(x)+\frac{1}{2}u^{\mathsf{T}}R(x)u\right)+\frac{\partial V(x,t)}{\partial x}^{\mathsf{T}}\left(A(x)+B(x)u\right)\right) \tag{2.47}$$

is a partial differential equation that is central to the dynamic programming. Here, the solution

$$V(x,t):\ =\min_{u_{[t,t_{\mathrm{f}}]}}\int_{t}^{t_{\mathrm{f}}}\left(Q\left(x(\tau)\right)+\frac{1}{2}u(\tau)^{\mathsf{T}}R\left(x(\tau)\right)u(\tau)\right)\mathrm{d}\tau \tag{2.48}$$

is called as the value function, where the notation $u_{[t,t_{\mathrm{f}}]}$ indicates that the control u is restricted to the interval $[t,t_{\mathrm{f}}]$. The Hamilton–Jacobi–Bellman equation is a necessary and sufficient condition for the optimality when it is solved over the whole state space [18].

Though we introduce the generating function based on the Hamiltonian system via the Pontryagin's minimum principle, it should also have connections with the value function. This is presented in the following theorem.

Theorem 2.5 ([36]). *For Problem 2.1, the relation between the value function and the generating function is*[†]

$$V(x,t)=F_{1\mathrm{b}}(x,x(t_{\mathrm{f}}),t) \tag{2.49}$$

where the value function $V(x,t)$ satisfies the Hamilton–Jacobi–Bellman equation (2.47) ($\forall t\in[t_0,t_{\mathrm{f}})$ and $\forall x\in\mathbb{R}^n$) and the terminal condition $V(x,t)|_{t=t_{\mathrm{f}}}=0$ on $x(t_{\mathrm{f}})=x_{\mathrm{term}}$.

Remark 2.2. According to the Legendre transformation (2.15), the relation (2.49) in Theorem 2.5 can be rewritten as

$$V(x,t)=F_{2\mathrm{b}}(x,\lambda(t_{\mathrm{f}}),t)-\lambda(t_{\mathrm{f}})^{\mathsf{T}}x(t_{\mathrm{f}}). \tag{2.50}$$

At time $t=t_{\mathrm{f}}$, we have $V(x,t)|_{t=t_{\mathrm{f}}}=(F_{2\mathrm{b}}(x,\lambda(t_{\mathrm{f}}),t)-\lambda(t_{\mathrm{f}})^{\mathsf{T}}x(t_{\mathrm{f}}))|_{t=t_{\mathrm{f}}}=0$ (according to Remark 2.1) which verifies the terminal condition presented in Theorem 2.5.

2.1.5 LQ optimal control problem

If we reduce Problem 2.1 into the LQ case, it can clearly exhibit the feature and advantage of the generating function method. This will be exhibited in this subsection.

First, consider the following continuous-time LQ optimal control problem.

[†]The generating function here is defined as $F_{1\mathrm{b}}(x,x(t_{\mathrm{f}}),t):\ =\int_{t}^{t_{\mathrm{f}}}(H(x(\tau),\lambda(\tau))-\lambda(\tau)^{\mathsf{T}}x(\tau))\mathrm{d}\tau$, while in [36], the generating function is defined as $F_{1\mathrm{b}}(x,x(t_{\mathrm{f}}),t):\ =-\int_{t}^{t_{\mathrm{f}}}(H(x(\tau),\lambda(\tau))-\lambda(\tau)^{\mathsf{T}}x(\tau))\mathrm{d}\tau$, so the relation here is negative to the original version.

Problem 2.2.

$$\min_{u} \int_{t_0}^{t_\mathrm{f}} \frac{1}{2}\left(x(t)^\mathsf{T} Q x(t) + u(t)^\mathsf{T} R u(t)\right) \mathrm{d}t \tag{2.51}$$

$$\text{s.t. } \dot{x}(t) = Ax(t) + Bu(t), \quad t \in [t_0, t_\mathrm{f}] \tag{2.52}$$

$$x(t_0) = x_\mathrm{init}, \quad x(t_\mathrm{f}) = x_\mathrm{term} \tag{2.53}$$

where the constant matrices $Q \in \mathbb{R}^{n\times n}$, $R \in \mathbb{R}^{m\times m}$, $A \in \mathbb{R}^{n\times n}$, and $B \in \mathbb{R}^{n\times m}$. Moreover, the matrices $Q \succcurlyeq 0$ and $R \succ 0$.

For the LQ case, we can get the exact expressions of the generating functions by solving the corresponding Hamilton–Jacobi equations. This is presented in the following proposition (only well-defined generating functions $F_{2\mathrm{f}}$, $F_{3\mathrm{f}}$, $F_{2\mathrm{b}}$, and $F_{3\mathrm{b}}$).

Proposition 2.2 ([39, 38])**.** *For Problem 2.2*

(i) *The generating function $F_{2\mathrm{f}}(x, \lambda(t_0), t)$ has the expression of*

$$F_{2\mathrm{f}}(x, \lambda(t_0), t) = \frac{1}{2} x^\mathsf{T} \mathscr{U}_{2\mathrm{f}}(t) x + \lambda(t_0)^\mathsf{T} \mathscr{V}_{2\mathrm{f}}(t) x + \frac{1}{2} \lambda(t_0)^\mathsf{T} \mathscr{W}_{2\mathrm{f}}(t) \lambda(t_0) \tag{2.54}$$

where the time-varying coefficients $\mathscr{U}_{2\mathrm{f}}(t) = \mathscr{U}_{2\mathrm{f}}(t)^\mathsf{T} \in \mathbb{R}^{n\times n}$, $\mathscr{V}_{2\mathrm{f}}(t) \in \mathbb{R}^{n\times n}$, and $\mathscr{W}_{2\mathrm{f}}(t) = \mathscr{W}_{2\mathrm{f}}(t)^\mathsf{T} \in \mathbb{R}^{n\times n}$ are the solutions of the ordinary differential equations ($t \in [t_0, t_\mathrm{f}]$)

$$\dot{\mathscr{U}}_{2\mathrm{f}}(t) = -\mathscr{U}_{2\mathrm{f}}(t)^\mathsf{T} A - A^\mathsf{T} \mathscr{U}_{2\mathrm{f}}(t) + \mathscr{U}_{2\mathrm{f}}(t)^\mathsf{T} G \mathscr{U}_{2\mathrm{f}}(t) - Q \tag{2.55}$$

$$\dot{\mathscr{V}}_{2\mathrm{f}}(t) = \mathscr{V}_{2\mathrm{f}}(t) G \mathscr{U}_{2\mathrm{f}}(t) - \mathscr{V}_{2\mathrm{f}}(t) A \tag{2.56}$$

$$\dot{\mathscr{W}}_{2\mathrm{f}}(t) = \mathscr{V}_{2\mathrm{f}}(t) G \mathscr{V}_{2\mathrm{f}}(t)^\mathsf{T} \tag{2.57}$$

with boundary conditions $\mathscr{U}_{2\mathrm{f}}(t_0) = 0$, $\mathscr{V}_{2\mathrm{f}}(t_0) = I$, and $\mathscr{W}_{2\mathrm{f}}(t_0) = 0$.

(ii) *The generating function $F_{3\mathrm{f}}(\lambda, x(t_0), t)$ has the expression of*

$$F_{3\mathrm{f}}(\lambda, x(t_0), t) = \frac{1}{2} \lambda^\mathsf{T} \mathscr{U}_{3\mathrm{f}}(t) \lambda + x(t_0)^\mathsf{T} \mathscr{V}_{3\mathrm{f}}(t) \lambda + \frac{1}{2} x(t_0)^\mathsf{T} \mathscr{W}_{3\mathrm{f}}(t) x(t_0) \tag{2.58}$$

where the time-varying coefficients $\mathscr{U}_{3\mathrm{f}}(t) = \mathscr{U}_{3\mathrm{f}}(t)^\mathsf{T} \in \mathbb{R}^{n\times n}$, $\mathscr{V}_{3\mathrm{f}}(t) \in \mathbb{R}^{n\times n}$, and $\mathscr{W}_{3\mathrm{f}}(t) = \mathscr{W}_{3\mathrm{f}}(t)^\mathsf{T} \in \mathbb{R}^{n\times n}$ are the solutions of the ordinary differential equations ($t \in [t_0, t_\mathrm{f}]$)

$$\dot{\mathscr{U}}_{3\mathrm{f}}(t) = A \mathscr{U}_{3\mathrm{f}}(t) - \mathscr{U}_{3\mathrm{f}}(t)^\mathsf{T} A^\mathsf{T} - \mathscr{U}_{3\mathrm{f}}(t)^\mathsf{T} Q \mathscr{U}_{3\mathrm{f}}(t) + G \tag{2.59}$$

$$\dot{\mathscr{V}}_{3\mathrm{f}}(t) = -\mathscr{V}_{3\mathrm{f}}(t) Q \mathscr{U}_{3\mathrm{f}}(t) + \mathscr{V}_{3\mathrm{f}}(t) A^\mathsf{T} \tag{2.60}$$

$$\dot{\mathscr{W}}_{3\mathrm{f}}(t) = -\mathscr{V}_{3\mathrm{f}}(t) Q \mathscr{V}_{3\mathrm{f}}(t)^\mathsf{T} \tag{2.61}$$

with boundary conditions $\mathscr{U}_{3\mathrm{f}}(t_0) = 0$, $\mathscr{V}_{3\mathrm{f}}(t_0) = -I$, and $\mathscr{W}_{3\mathrm{f}}(t_0) = 0$.

(iii) *The generating function* $F_{2b}(x, \lambda(t_f), t)$ *has the expression of*

$$F_{2b}(x, \lambda(t_f), t) = \frac{1}{2}x^{\mathsf{T}}\mathscr{U}_{2b}(t)x + \lambda(t_f)^{\mathsf{T}}\mathscr{V}_{2b}(t)x + \frac{1}{2}\lambda(t_f)^{\mathsf{T}}\mathscr{W}_{2b}(t)\lambda(t_f) \tag{2.62}$$

where the time-varying coefficients $\mathscr{U}_{2b}(t) = \mathscr{U}_{2b}(t)^{\mathsf{T}} \in \mathbb{R}^{n\times n}$, $\mathscr{V}_{2b}(t) \in \mathbb{R}^{n\times n}$, *and* $\mathscr{W}_{2b}(t) = \mathscr{W}_{2b}(t)^{\mathsf{T}} \in \mathbb{R}^{n\times n}$ *are the solutions of the ordinary differential equations* ($t \in [t_0, t_f]$)

$$\dot{\mathscr{U}}_{2b}(t) = -\mathscr{U}_{2b}(t)^{\mathsf{T}}A - A^{\mathsf{T}}\mathscr{U}_{2b}(t) + \mathscr{U}_{2b}(t)^{\mathsf{T}}G\mathscr{U}_{2b}(t) - Q \tag{2.63}$$

$$\dot{\mathscr{V}}_{2b}(t) = \mathscr{V}_{2b}(t)G\mathscr{U}_{2b}(t) - \mathscr{V}_{2b}(t)A \tag{2.64}$$

$$\dot{\mathscr{W}}_{2b}(t) = \mathscr{V}_{2b}(t)G\mathscr{V}_{2b}(t)^{\mathsf{T}} \tag{2.65}$$

with boundary conditions $\mathscr{U}_{2b}(t_f) = 0$, $\mathscr{V}_{2b}(t_f) = I$, *and* $\mathscr{W}_{2b}(t_f) = 0$.

(iv) *The generating function* $F_{3b}(\lambda, x(t_f), t)$ *has the expression of*

$$F_{3b}(\lambda, x(t_f), t) = \frac{1}{2}\lambda^{\mathsf{T}}\mathscr{U}_{3b}(t)\lambda + x(t_f)^{\mathsf{T}}\mathscr{V}_{3b}(t)\lambda + \frac{1}{2}x(t_f)^{\mathsf{T}}\mathscr{W}_{3b}(t)x(t_f) \tag{2.66}$$

where the time-varying coefficients $\mathscr{U}_{3b}(t) = \mathscr{U}_{3b}(t)^{\mathsf{T}} \in \mathbb{R}^{n\times n}$, $\mathscr{V}_{3b}(t) \in \mathbb{R}^{n\times n}$, *and* $\mathscr{W}_{3b}(t) = \mathscr{W}_{3b}(t)^{\mathsf{T}} \in \mathbb{R}^{n\times n}$ *are the solutions of the ordinary differential equations* ($t \in [t_0, t_f]$)

$$\dot{\mathscr{U}}_{3b}(t) = A\mathscr{U}_{3b}(t) - \mathscr{U}_{3b}(t)^{\mathsf{T}}A^{\mathsf{T}} - \mathscr{U}_{3b}(t)^{\mathsf{T}}Q\mathscr{U}_{3b}(t) + G \tag{2.67}$$

$$\dot{\mathscr{V}}_{3b}(t) = -\mathscr{V}_{3b}(t)Q\mathscr{U}_{3b}(t) + \mathscr{V}_{3b}(t)A^{\mathsf{T}} \tag{2.68}$$

$$\dot{\mathscr{W}}_{3b}(t) = -\mathscr{V}_{3b}(t)Q\mathscr{V}_{3b}(t)^{\mathsf{T}} \tag{2.69}$$

with boundary conditions $\mathscr{U}_{3b}(t_f) = 0$, $\mathscr{V}_{3b}(t_f) = -I$, *and* $\mathscr{W}_{3b}(t_f) = 0$.

Here, $G := BR^{-1}B^{\mathsf{T}}$ *and* $I \in \mathbb{R}^{n\times n}$ *is the identity matrix.*

Remark 2.3. By substituting (2.62) into Theorem 2.3, we get the optimal solutions of Problem 2.2 by single generating function method [36] as

$$u^*(t) = -R^{-1}B^{\mathsf{T}}\left(\mathscr{U}_{2b}(t)x + \mathscr{V}_{2b}(t)^{\mathsf{T}}\lambda(t_f)\right), \quad t \in [t_0, t_f] \tag{2.70}$$

where

$$\lambda(t_f) = \mathscr{W}_{2b}(t_0)^{-1}\left(x(t_f) - \mathscr{V}_{2b}(t_0)x(t_0)\right). \tag{2.71}$$

Remark 2.4. By substituting (2.58) and (2.66) into Theorem 2.4, we get the optimal solutions of Problem 2.2 by double generating functions method [39] as

$$\begin{bmatrix} x^*(t) \\ u^*(t) \end{bmatrix} = \begin{bmatrix} \mathscr{U}_{3b}(t)\left(\mathscr{U}_{3f}(t) - \mathscr{U}_{3b}(t)\right)^{-1}\mathscr{V}_{3f}(t)^{\mathsf{T}}, \\ R^{-1}B^{\mathsf{T}}\left(\mathscr{U}_{3f}(t) - \mathscr{U}_{3b}(t)\right)^{-1}\mathscr{V}_{3f}(t)^{\mathsf{T}}, \\ -\mathscr{U}_{3f}(t)\left(\mathscr{U}_{3f}(t) - \mathscr{U}_{3b}(t)\right)^{-1}\mathscr{V}_{3b}(t)^{\mathsf{T}} \\ -R^{-1}B^{\mathsf{T}}\left(\mathscr{U}_{3f}(t) - \mathscr{U}_{3b}(t)\right)^{-1}\mathscr{V}_{3b}(t)^{\mathsf{T}} \end{bmatrix}^{\mathsf{T}} \begin{bmatrix} x(t_0) \\ x(t_f) \end{bmatrix}, \quad t \in [t_0, t_f] \tag{2.72}$$

Remark 2.5. Notice the inverse term $(\mathscr{U}_{3\mathrm{f}}(t) - \mathscr{U}_{3\mathrm{b}}(t))^{-1}$ in (2.72). It is necessary for us to analyze its invertibility. Besides (2.72), there have another five generators for optimal solutions constructed by selecting each two different generating functions among $F_{2\mathrm{f}}$, $F_{3\mathrm{f}}$, $F_{2\mathrm{b}}$, and $F_{3\mathrm{b}}$. It is proven in [38] that the terms in the generators constructed by double generating functions with same time directions will become singular when the time goes infinity which may cause instability in numerical computations. Therefore, when we select optimal generators, the ones constructed by the pair of generating functions with same time direction should be avoided.

2.2 Discrete-time case

Consider the following discrete-time optimal control problem.

Problem 2.3.

$$\min_{u} \sum_{k=0}^{N-1} \left(Q(x_k) + \frac{1}{2} u_k^{\mathsf{T}} R(x_k) u_k \right) \tag{2.73}$$

$$\text{s.t. } x_{k+1} = A(x_k) + B(x_k)u_k, \quad k = 0, 1, \cdots, N-1 \tag{2.74}$$

$$x_0 = x_{\mathrm{init}}, \quad x_N = x_{\mathrm{term}} \tag{2.75}$$

where k is the time step, $x_k \in \mathbb{R}^n$ and $u_k \in \mathbb{R}^m$ are the state and input variables, respectively. Functions $Q\colon \mathbb{R}^n \to \mathbb{R}$, $R\colon \mathbb{R}^n \to \mathbb{R}^{m\times m}$, $A\colon \mathbb{R}^n \to \mathbb{R}^n$, and $B\colon \mathbb{R}^n \to \mathbb{R}^{n\times m}$. Moreover, the function $Q \succcurlyeq 0$, and the matrix $R(x_k) \succ 0, \forall x_k \in \mathbb{R}^n$. In (2.75), $x_{\mathrm{init}} \in \mathbb{R}^n$ and $x_{\mathrm{term}} \in \mathbb{R}^n$ are the given initial and terminal state values, respectively.

2.2.1 Necessary conditions for optimality

As introduced in [48], the first-order necessary conditions for minimizing Problem 2.3 can be represented by the right Hamiltonian or the left Hamiltonian.

We first introduce the right one which is more general. The right pre-Hamiltonian $\bar{H}^+\colon \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ is given by adjoining the right hand side of the difference equation in (2.74) to the cost in (2.73) as

$$\bar{H}^+(x_k, \lambda_{k+1}, u_k) \colon = \left(Q(x_k) + \frac{1}{2} u_k^{\mathsf{T}} R(x_k) u_k \right) + \lambda_{k+1}^{\mathsf{T}} \left(A(x_k) + B(x_k)u_k \right) \tag{2.76}$$

where $\lambda_{k+1} \in \mathbb{R}^n$ is introduced as the costate. We denote the optimal state, costate, and input of Problem 2.3 as x_k^*, λ_k^*, and u_k^*, respectively.

Now, we give the first-order necessary conditions represented by the right Hamiltonian.

Theorem 2.6 ([48]). *For Problem 2.3, there is a vector function λ_k^* such that the triple $(x_k^*, \lambda_k^*, u_k^*)$ satisfies ($k = 0, 1, \cdots, N-1$)*

$$x_{k+1} = \frac{\partial \bar{H}^+(x_k, \lambda_{k+1}, u_k)}{\partial \lambda_{k+1}}, \qquad x_0 = x_{\mathrm{init}}, \quad x_N = x_{\mathrm{term}} \tag{2.77}$$

$$\lambda_k = \frac{\partial \bar{H}^+(x_k, \lambda_{k+1}, u_k)}{\partial x_k} \tag{2.78}$$

$$u_k = \arg\min_{\bar{u}_k} \bar{H}^+(x_k, \lambda_{k+1}, \bar{u}_k) \equiv -R(x_k)^{-1}B(x_k)^{\mathsf{T}}\lambda_{k+1}. \tag{2.79}$$

Substitution of (2.79) into the right pre-Hamiltonian (2.76) and the Hamilton's equations (2.77)–(2.78) gives the right Hamiltonian system for the state and costate

$$H^+(x_k, \lambda_{k+1}) = Q(x_k) + \lambda_{k+1}^{\mathsf{T}}A(x_k) - \frac{1}{2}\lambda_{k+1}^{\mathsf{T}}B(x_k)R(x_k)^{-1}B(x_k)^{\mathsf{T}}\lambda_{k+1} \tag{2.80}$$

$$x_{k+1} = \frac{\partial H^+(x_k, \lambda_{k+1})}{\partial \lambda_{k+1}} \tag{2.81}$$

$$\lambda_k = \frac{\partial H^+(x_k, \lambda_{k+1})}{\partial x_k}. \tag{2.82}$$

The first-order necessary conditions can also be represented by the left Hamiltonian $\bar{H}^- : \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$, which is presented in the following theorem.

Theorem 2.7 ([48]). *For Problem 2.3, there is a vector function λ_k^* such that the triple $(x_k^*, \lambda_k^*, u_k^*)$ satisfies ($k = 0, 1, \cdots, N-1$)*

$$x_k = -\frac{\partial \bar{H}^-(\lambda_k, x_{k+1}, u_k)}{\partial \lambda_k}, \qquad x_0 = x_{\text{init}}, \quad x_N = x_{\text{term}} \tag{2.83}$$

$$\lambda_{k+1} = -\frac{\partial \bar{H}^-(\lambda_k, x_{k+1}, u_k)}{\partial x_{k+1}} \tag{2.84}$$

$$u_k = \arg\min_{\bar{u}_k} \bar{H}^-(\lambda_k, x_{k+1}, \bar{u}_k). \tag{2.85}$$

Unlike the right pre-Hamiltonian, the left pre-Hamiltonian here is unknown, so we can not give the exact right Hamiltonian system by the substitution of (2.85). However, the exact left Hamiltonian can be obtained through Legendre transformation from the right Hamiltonian [48]

$$H^-(\lambda_k, x_{k+1}) = H^+(x_k, \lambda_{k+1}) - \lambda_k^{\mathsf{T}}x_k - \lambda_{k+1}^{\mathsf{T}}x_{k+1}. \tag{2.86}$$

Based on this, we can give the left Hamiltonian system for the state and costate

$$x_k = -\frac{\partial H^-(\lambda_k, x_{k+1})}{\partial \lambda_k} \tag{2.87}$$

$$\lambda_{k+1} = -\frac{\partial H^-(\lambda_k, x_{k+1})}{\partial x_{k+1}}. \tag{2.88}$$

For nonlinear problems, the exact left Hamiltonian can hardly be found by the above Legendre transformation (2.86), so the right Hamiltonian is of priority.

2.2.2 Hamilton–Jacobi equation and generating function

Similar as the continuous-time case, there also have forward and backward generating functions in the discrete-time case. Each of them also has four kinds of functions by selecting each two different variables from the current state-costate and the boundary state-costate. Totally, there are eight kinds of discrete generating functions $F_{1\text{f}}(x_k, x_0, k)$, $F_{2\text{f}}(x_k, \lambda_0, k)$, $F_{3\text{f}}(\lambda_k, x_0, k)$, $F_{4\text{f}}(\lambda_k, \lambda_0, k)$, $F_{1\text{b}}(x_k, x_N, k)$, $F_{2\text{b}}(x_k, \lambda_N, k)$, $F_{3\text{b}}(\lambda_k, x_N, k)$, and $F_{4\text{b}}(\lambda_k, \lambda_N, k)$.

Similarly, there also exist Legendre transformations [49] between forward generating functions

$$F_{1\mathrm{f}}(x_k, x_0, k) = F_{2\mathrm{f}}(x_k, \lambda_0, k) - \lambda_0^{\mathsf{T}} x_0 \tag{2.89}$$
$$F_{1\mathrm{f}}(x_k, x_0, k) = F_{3\mathrm{f}}(\lambda_k, x_0, k) + \lambda_k^{\mathsf{T}} x_k \tag{2.90}$$
$$F_{1\mathrm{f}}(x_k, x_0, k) = F_{4\mathrm{f}}(\lambda_k, \lambda_0, k) + \lambda_k^{\mathsf{T}} x_k - \lambda_0^{\mathsf{T}} x_0 \tag{2.91}$$

and between backward generating functions

$$F_{1\mathrm{b}}(x_k, x_N, k) = F_{2\mathrm{b}}(x_k, \lambda_N, k) - \lambda_N^{\mathsf{T}} x_N \tag{2.92}$$
$$F_{1\mathrm{b}}(x_k, x_N, k) = F_{3\mathrm{b}}(\lambda_k, x_N, k) + \lambda_k^{\mathsf{T}} x_k \tag{2.93}$$
$$F_{1\mathrm{b}}(x_k, x_N, k) = F_{4\mathrm{b}}(\lambda_k, \lambda_N, k) + \lambda_k^{\mathsf{T}} x_k - \lambda_N^{\mathsf{T}} x_N. \tag{2.94}$$

Then, we give the basic relations and Hamilton–Jacobi equations for the eight kinds of discrete generating functions in the following proposition.

Proposition 2.3 ([50]). *For Problem 2.3*

(i) *The generating function $F_{1\mathrm{f}}(x_k, x_0, k)$ satisfying the Hamilton–Jacobi equation*

$$\begin{aligned} F_{1\mathrm{f}}(x_{k-1}, x_0, k-1) =& F_{1\mathrm{f}}(x_k, x_0, k) - \left(\frac{\partial F_{1\mathrm{f}}(x_k, x_0, k)}{\partial x_k}\right)^{\mathsf{T}} x_k \\ &+ H^{+}\left(x_{k-1}, \frac{\partial F_{1\mathrm{f}}(x_k, x_0, k)}{\partial x_k}\right) \end{aligned} \tag{2.95}$$

specifies the family of forward canonical transformations $(x_0, \lambda_0) \mapsto (x_k, \lambda_k)$, $k = 0, 1, \cdots, N$, by the basic relations

$$\lambda_k = \frac{\partial F_{1\mathrm{f}}(x_k, x_0, k)}{\partial x_k} \tag{2.96}$$
$$\lambda_0 = -\frac{\partial F_{1\mathrm{f}}(x_k, x_0, k)}{\partial x_0}. \tag{2.97}$$

(ii) *The generating function $F_{2\mathrm{f}}(x_k, \lambda_0, k)$ satisfying the Hamilton–Jacobi equation*

$$\begin{aligned} F_{2\mathrm{f}}(x_{k-1}, \lambda_0, k-1) =& F_{2\mathrm{f}}(x_k, \lambda_0, k) - \left(\frac{\partial F_{2\mathrm{f}}(x_k, \lambda_0, k)}{\partial x_k}\right)^{\mathsf{T}} x_k \\ &+ H^{+}\left(x_{k-1}, \frac{\partial F_{2\mathrm{f}}(x_k, \lambda_0, k)}{\partial x_k}\right) \end{aligned} \tag{2.98}$$

specifies the family of forward canonical transformations $(x_0, \lambda_0) \mapsto (x_k, \lambda_k)$, $k = 0, 1, \cdots, N$, by the basic relations

$$\lambda_k = \frac{\partial F_{2\mathrm{f}}(x_k, \lambda_0, k)}{\partial x_k} \tag{2.99}$$
$$x_0 = \frac{\partial F_{2\mathrm{f}}(x_k, \lambda_0, k)}{\partial \lambda_0}. \tag{2.100}$$

(iii) *The generating function $F_{3\mathrm{f}}(\lambda_k, x_0, k)$ satisfying the Hamilton–Jacobi equation*

$$\begin{aligned} F_{3\mathrm{f}}(\lambda_{k+1}, x_0, k+1) =& F_{3\mathrm{f}}(\lambda_k, x_0, k) - \lambda_k^{\mathsf{T}} \left(\frac{\partial F_{3\mathrm{f}}(\lambda_k, x_0, k)}{\partial \lambda_k} \right) \\ & - H^+ \left(-\frac{\partial F_{3\mathrm{f}}(\lambda_k, x_0, k)}{\partial \lambda_k}, \lambda_{k+1} \right) \end{aligned} \tag{2.101}$$

specifies the family of forward canonical transformations $(x_0, \lambda_0) \mapsto (x_k, \lambda_k)$, $k = 0, 1, \cdots, N$, by the basic relations

$$x_k = -\frac{\partial F_{3\mathrm{f}}(\lambda_k, x_0, k)}{\partial \lambda_k} \tag{2.102}$$

$$\lambda_0 = -\frac{\partial F_{3\mathrm{f}}(\lambda_k, x_0, k)}{\partial x_0}. \tag{2.103}$$

(iv) *The generating function $F_{4\mathrm{f}}(\lambda_k, \lambda_0, k)$ satisfying the Hamilton–Jacobi equation*

$$\begin{aligned} F_{4\mathrm{f}}(\lambda_{k+1}, \lambda_0, k+1) =& F_{4\mathrm{f}}(\lambda_k, \lambda_0, k) - \lambda_k^{\mathsf{T}} \left(\frac{\partial F_{4\mathrm{f}}(\lambda_k, \lambda_0, k)}{\partial \lambda_k} \right) \\ & - H^+ \left(-\frac{\partial F_{4\mathrm{f}}(\lambda_k, \lambda_0, k)}{\partial \lambda_k}, \lambda_{k+1} \right) \end{aligned} \tag{2.104}$$

specifies the family of forward canonical transformations $(x_0, \lambda_0) \mapsto (x_k, \lambda_k)$, $k = 0, 1, \cdots, N$, by the basic relations

$$x = -\frac{\partial F_{4\mathrm{f}}(\lambda_k, \lambda_0, k)}{\partial \lambda} \tag{2.105}$$

$$x_0 = \frac{\partial F_{4\mathrm{f}}(\lambda_k, \lambda_0, k)}{\partial \lambda_0}. \tag{2.106}$$

(v) *The generating function $F_{1\mathrm{b}}(x_k, x_N, k)$ satisfying the Hamilton–Jacobi equation*

$$\begin{aligned} F_{1\mathrm{b}}(x_{k-1}, x_N, k-1) =& F_{1\mathrm{b}}(x_k, x_N, k) - \left(\frac{\partial F_{1\mathrm{b}}(x_k, x_N, k)}{\partial x_k} \right)^{\mathsf{T}} x_k \\ & + H^+ \left(x_{k-1}, \frac{\partial F_{1\mathrm{b}}(x_k, x_N, k)}{\partial x_k} \right) \end{aligned} \tag{2.107}$$

specifies the family of backward canonical transformations $(x_N, \lambda_N) \mapsto (x_k, \lambda_k)$, $k = 0, 1, \cdots, N$, by the basic relations

$$\lambda_k = \frac{\partial F_{1\mathrm{b}}(x_k, x_N, k)}{\partial x_k} \tag{2.108}$$

$$\lambda_N = -\frac{\partial F_{1\mathrm{b}}(x_k, x_N, k)}{\partial x_N}. \tag{2.109}$$

(vi) *The generating function* $F_{2\mathrm{b}}(x_k, \lambda_N, k)$ *satisfying the Hamilton–Jacobi equation*

$$\begin{aligned} F_{2\mathrm{b}}(x_{k-1}, \lambda_N, k-1) =& F_{2\mathrm{b}}(x_k, \lambda_N, k) - \left(\frac{\partial F_{2\mathrm{b}}(x_k, \lambda_N, k)}{\partial x_k}\right)^{\mathsf{T}} x_k \\ &+ H^+\left(x_{k-1}, \frac{\partial F_{2\mathrm{b}}(x_k, \lambda_N, k)}{\partial x_k}\right) \end{aligned} \tag{2.110}$$

specifies the family of backward canonical transformations $(x_N, \lambda_N) \mapsto (x_k, \lambda_k)$, $k = 0, 1, \cdots, N$, *by the basic relations*

$$\lambda_k = \frac{\partial F_{2\mathrm{b}}(x_k, \lambda_N, k)}{\partial x_k} \tag{2.111}$$

$$x_N = \frac{\partial F_{2\mathrm{b}}(x_k, \lambda_N, k)}{\partial \lambda_N}. \tag{2.112}$$

(vii) *The generating function* $F_{3\mathrm{b}}(\lambda_k, x_N, k)$ *satisfying the Hamilton–Jacobi equation*

$$\begin{aligned} F_{3\mathrm{b}}(\lambda_{k+1}, x_N, k+1) =& F_{3\mathrm{b}}(\lambda_k, x_N, k) - \lambda_k^{\mathsf{T}}\left(\frac{\partial F_{3\mathrm{b}}(\lambda_k, x_N, k)}{\partial \lambda_k}\right) \\ &- H^+\left(-\frac{\partial F_{3\mathrm{b}}(\lambda_k, x_N, k)}{\partial \lambda_k}, \lambda_{k+1}\right) \end{aligned} \tag{2.113}$$

specifies the family of backward canonical transformations $(x_N, \lambda_N) \mapsto (x_k, \lambda_k)$, $k = 0, 1, \cdots, N$, *by the basic relations*

$$x_k = -\frac{\partial F_{3\mathrm{b}}(\lambda_k, x_N, k)}{\partial \lambda_k} \tag{2.114}$$

$$\lambda_N = -\frac{\partial F_{3\mathrm{b}}(\lambda_k, x_N, k)}{\partial x_N}. \tag{2.115}$$

(viii) *The generating function* $F_{4\mathrm{b}}(\lambda_k, \lambda_N, k)$ *satisfying the Hamilton–Jacobi equation*

$$\begin{aligned} F_{4\mathrm{b}}(\lambda_{k+1}, \lambda_N, k+1) =& F_{4\mathrm{b}}(\lambda_k, \lambda_N, k) - \lambda_k^{\mathsf{T}}\left(\frac{\partial F_{4\mathrm{b}}(\lambda_k, \lambda_N, k)}{\partial \lambda_k}\right) \\ &- H^+\left(-\frac{\partial F_{4\mathrm{b}}(\lambda_k, \lambda_N, k)}{\partial \lambda_k}, \lambda_{k+1}\right) \end{aligned} \tag{2.116}$$

specifies the family of backward canonical transformations $(x_N, \lambda_N) \mapsto (x_k, \lambda_k)$, $k = 0, 1, \cdots, N$, *by the basic relations*

$$x = -\frac{\partial F_{4\mathrm{b}}(\lambda_k, \lambda_N, k)}{\partial \lambda} \tag{2.117}$$

$$x_N = \frac{\partial F_{4\mathrm{b}}(\lambda_k, \lambda_N, k)}{\partial \lambda_N}. \tag{2.118}$$

Remark 2.6. Note that all the Hamilton–Jacobi equations in the above proposition are represented in terms of the right Hamiltonian. We can also write these Hamilton–Jacobi equations in terms of the left Hamiltonian by the substitution of the Legendre transformation [48]

$$H^+(x_k,\lambda_{k+1}) = H^-(\lambda_k,x_{k+1}) + \lambda_k^{\mathsf{T}} x_k + \lambda_{k+1}^{\mathsf{T}} x_{k+1}. \tag{2.119}$$

into (2.95), (2.98), (2.101), (2.104), (2.107), (2.110), (2.113), and (2.116) to get

$$F_{1\mathrm{f}}(x_{k+1},x_0,k+1) = F_{1\mathrm{f}}(x_k,x_0,k) - \left(\frac{\partial F_{1\mathrm{f}}(x_k,x_0,k)}{\partial x_k}\right)^{\mathsf{T}} x_k + H^-\left(\frac{\partial F_{1\mathrm{f}}(x_k,x_0,k)}{\partial x_k}, x_{k+1}\right) \tag{2.120}$$

$$F_{2\mathrm{f}}(x_{k+1},\lambda_0,k+1) = F_{2\mathrm{f}}(x_k,\lambda_0,k) - \left(\frac{\partial F_{2\mathrm{f}}(x_k,\lambda_0,k)}{\partial x_k}\right)^{\mathsf{T}} x_k - H^-\left(\frac{\partial F_{2\mathrm{f}}(x_k,\lambda_0,k)}{\partial x_k}, x_{k+1}\right) \tag{2.121}$$

$$F_{3\mathrm{f}}(\lambda_{k-1},x_0,k-1) = F_{3\mathrm{f}}(\lambda_k,x_0,k) - \lambda_k^{\mathsf{T}}\left(\frac{\partial F_{3\mathrm{f}}(\lambda_k,x_0,k)}{\partial \lambda_k}\right) - H^-\left(\lambda_{k-1}, -\frac{\partial F_{3\mathrm{f}}(\lambda_k,x_0,k)}{\partial \lambda_k}\right) \tag{2.122}$$

$$F_{4\mathrm{f}}(\lambda_{k-1},\lambda_0,k-1) = F_{4\mathrm{f}}(\lambda_k,\lambda_0,k) - \lambda_k^{\mathsf{T}}\left(\frac{\partial F_{4\mathrm{f}}(\lambda_k,\lambda_0,k)}{\partial \lambda_k}\right) + H^-\left(\lambda_{k-1}, -\frac{\partial F_{4\mathrm{f}}(\lambda_k,\lambda_0,k)}{\partial \lambda_k}\right) \tag{2.123}$$

$$F_{1\mathrm{b}}(x_{k+1},x_N,k+1) = F_{1\mathrm{b}}(x_k,x_N,k) - \left(\frac{\partial F_{1\mathrm{b}}(x_k,x_N,k)}{\partial x_k}\right)^{\mathsf{T}} x_k + H^-\left(\frac{\partial F_{1\mathrm{b}}(x_k,x_N,k)}{\partial x_k}, x_{k+1}\right) \tag{2.124}$$

$$F_{2\mathrm{b}}(x_{k+1},\lambda_N,k+1) = F_{2\mathrm{b}}(x_k,\lambda_N,k) - \left(\frac{\partial F_{2\mathrm{b}}(x_k,\lambda_N,k)}{\partial x_k}\right)^{\mathsf{T}} x_k - H^-\left(\frac{\partial F_{2\mathrm{b}}(x_k,\lambda_N,k)}{\partial x_k}, x_{k+1}\right) \tag{2.125}$$

$$F_{3\mathrm{b}}(\lambda_{k-1},x_N,k-1) = F_{3\mathrm{b}}(\lambda_k,x_N,k) - \lambda_k^{\mathsf{T}}\left(\frac{\partial F_{3\mathrm{b}}(\lambda_k,x_N,k)}{\partial \lambda_k}\right) - H^-\left(\lambda_{k-1}, -\frac{\partial F_{3\mathrm{b}}(\lambda_k,x_N,k)}{\partial \lambda_k}\right) \tag{2.126}$$

$$F_{4\mathrm{b}}(\lambda_{k-1},\lambda_N,k-1) = F_{4\mathrm{b}}(\lambda_k,\lambda_N,k) - \lambda_k^{\mathsf{T}}\left(\frac{\partial F_{4\mathrm{b}}(\lambda_k,\lambda_N,k)}{\partial \lambda_k}\right) + H^-\left(\lambda_{k-1}, -\frac{\partial F_{4\mathrm{b}}(\lambda_k,\lambda_N,k)}{\partial \lambda_k}\right). \tag{2.127}$$

Remark 2.7. Similar as the continuous generating functions, the discrete generating functions $F_{1\mathrm{f}}$ and $F_{4\mathrm{f}}$ are not well-defined at initial time, and $F_{1\mathrm{b}}$ and $F_{4\mathrm{b}}$ are not well-defined at terminal time, while the discrete generating functions $F_{2\mathrm{f}}$, $F_{3\mathrm{f}}$, $F_{2\mathrm{b}}$, and $F_{3\mathrm{b}}$ are all well-defined at boundary times. In summary, we have $F_{2\mathrm{f}}(x_k,\lambda_0,k)|_{k=0} = \lambda_0^{\mathsf{T}} x_0$, $F_{3\mathrm{f}}(\lambda_k,x_0,k)|_{k=0} = -\lambda_0^{\mathsf{T}} x_0$, $F_{2\mathrm{b}}(x_k,\lambda_N,k)|_{k=N} = \lambda_N^{\mathsf{T}} x_N$, and $F_{3\mathrm{b}}(\lambda_k,x_N,k)|_{k=N} = -\lambda_N^{\mathsf{T}} x_N$.

2.2.3 Relation between generating function and value function

The Bellman equation

$$V(x_k,k) = \min_{u_k}\left(\left(Q(x_k) + \frac{1}{2}u_k^{\mathsf{T}} R(x_k) u_k\right) + V\left(A(x_k) + B(x_k)u_k, k+1\right)\right) \tag{2.128}$$

is central to the discrete dynamic programming. Here, the solution

$$V(x_k,k) \colon = \min_{u_{[k,N-1]}} \sum_{i=k}^{N-1}\left(Q(x_i) + \frac{1}{2}u_i^{\mathsf{T}} R(x_i) u_i\right) \tag{2.129}$$

is called as the value function, where the notation $u_{[k,N-1]}$ indicates that the discrete control u_k is restricted to the interval $[k, N-1]$.

Similar as the continuous-time case, there also have relations between the generating function and the value function. This is presented in the following theorem.

Theorem 2.8. *For Problem 2.3, the relation between the value function and the generating function is*

$$V(x_k,k)=F_{1\mathrm{b}}(x_k,x_N,k) \tag{2.130}$$

where the value function $V(x_k,k)$ satisfies the Bellman equation (2.128) ($\forall k=0,1,\cdots,N-1$ and $\forall x_k\in\mathbb{R}^n$) and the terminal condition $V(x_k,k)|_{k=N}=0$ on $x_N=x_{\mathrm{term}}$.

Proof. From (2.129), we have

$$\begin{aligned}
V(x_k,k):&=\min_{u_{[k,N-1]}}\sum_{i=k}^{N-1}\left(Q(x_i)+\frac{1}{2}u_i^{\mathsf{T}}R(x_i)u_i\right)\\
&=\min_{u_{[k,N-1]}}\sum_{i=k}^{N-1}\left(\left(Q(x_i)+\frac{1}{2}u_i^{\mathsf{T}}R(x_i)u_i\right)+\lambda_{i+1}^{\mathsf{T}}\Big(A(x_i)+B(x_i)u_i-x_{i+1}\Big)\right)\\
&=\min_{u_{[k,N-1]}}\sum_{i=k}^{N-1}\left(\bar{H}^+(x_i,\lambda_{i+1},u_i)-\lambda_{i+1}^{\mathsf{T}}x_{i+1}\right)\\
&=\sum_{i=k}^{N-1}\left(H^+(x_i,\lambda_{i+1})-\lambda_{i+1}^{\mathsf{T}}x_{i+1}\right).
\end{aligned}$$

According to [50], the generating function $F_{1\mathrm{b}}$ has the expression of

$$F_{1\mathrm{b}}(x_k,x_N,k)=\sum_{i=k}^{N-1}\left(H^+(x_i,\lambda_{i+1})-\lambda_{i+1}^{\mathsf{T}}x_{i+1}\right).$$

Therefore, we have

$$V(x_k,k)=F_{1\mathrm{b}}(x_k,x_N,k)$$

which is (2.130). Here, the value function $V(x_k,k)$ satisfies the Bellman equation (2.128) ($\forall k=0,1,\cdots,N-1$ and $\forall x_k\in\mathbb{R}^n$) and the terminal condition $V(x_k,k)|_{k=N}=0$ on $x_N=x_{\mathrm{term}}$. □

Remark 2.8. According to the Legendre transformation (2.92), the relation (2.130) in Theorem 2.8 can be rewritten as

$$V(x_k,k)=F_{2\mathrm{b}}(x_k,\lambda_N,k)-\lambda_N^{\mathsf{T}}x_N. \tag{2.131}$$

At time $k=N$, we have $V(x_k,k)|_{k=N}=(F_{2\mathrm{b}}(x_k,\lambda_N,N)-\lambda_N^{\mathsf{T}}x_N)|_{k=N}=0$ (according to Remark 2.7) which verifies the terminal condition presented in Theorem 2.8.

2.2.4 LQ optimal control problem

So far, for the case of discrete-time problems, the generating functions have only been applied to LQ optimal control, which we will introduce here.

First, consider the following discrete-time LQ optimal control problem.

Problem 2.4.

$$\min_{u} \sum_{k=0}^{N-1} \frac{1}{2} \left(x_k^{\mathsf{T}} Q x_k + u_k^{\mathsf{T}} R u_k \right) \tag{2.132}$$

$$\text{s.t. } x_{k+1} = A x_k + B u_k, \quad k = 0, 1, \cdots, N-1 \tag{2.133}$$

$$x_0 = x_{\text{init}}, \quad x_N = x_{\text{term}} \tag{2.134}$$

where the constant matrices $Q \in \mathbb{R}^{n\times n}$, $R \in \mathbb{R}^{m\times m}$, $A \in \mathbb{R}^{n\times n}$, and $B \in \mathbb{R}^{n\times m}$. Moreover, the matrices $Q \succcurlyeq 0$ and $R \succ 0$.

For the LQ case, we can get the explicit expressions of the generating functions by solving the corresponding Hamilton–Jacobi equations. The reference [50] only gives the expressions of $F_{1\text{b}}$ and $F_{2\text{b}}$. Since $F_{1\text{b}}$ is not well-defined at terminal time according to Remark 2.7, we here only present $F_{2\text{b}}$ in the following proposition.

Proposition 2.4 ([50])**.** *The generating function $F_{2\text{b}}(x_k, \lambda_N, k)$ for Problem 2.4 has the expression of*

$$F_{2\text{b}}(x_k, \lambda_N, k) = \frac{1}{2} x_k^{\mathsf{T}} \mathscr{U}_{2\text{b},k} x_k + \lambda_N^{\mathsf{T}} \mathscr{V}_{2\text{b},k} x_k + \frac{1}{2} \lambda_N^{\mathsf{T}} \mathscr{W}_{2\text{b},k} \lambda_N \tag{2.135}$$

where the coefficients $\mathscr{U}_{2\text{b},k} = \mathscr{U}_{2\text{b},k}^{\mathsf{T}} \in \mathbb{R}^{n\times n}$, $\mathscr{V}_{2\text{b},k} \in \mathbb{R}^{n\times n}$, and $\mathscr{W}_{2\text{b},k} = \mathscr{U}_{2\text{b},k}^{\mathsf{T}} \in \mathbb{R}^{n\times n}$ are the solutions of the difference equations ($k = N, N-1, \cdots, 1$)

$$\mathscr{U}_{2\text{b},k-1} = A^{\mathsf{T}} (I + \mathscr{U}_{2\text{b},k} G)^{-1} \mathscr{U}_{2\text{b},k} A + Q \tag{2.136}$$

$$\mathscr{V}_{2\text{b},k-1} = \mathscr{V}_{2\text{b},k} (I + G \mathscr{U}_{2\text{b},k})^{-1} A \tag{2.137}$$

$$\mathscr{W}_{2\text{b},k-1} = \mathscr{W}_{2\text{b},k} - \mathscr{V}_{2\text{b},k} (I + G \mathscr{U}_{2\text{b},k})^{-1} G \mathscr{V}_{2\text{b},k}^{\mathsf{T}} \tag{2.138}$$

with boundary conditions $\mathscr{U}_{2\text{f},N} = 0$, $\mathscr{V}_{2\text{f},N} = I$, and $\mathscr{W}_{2\text{f},N} = 0$.

Based on the generating function presented in Proposition 2.4, we can use it to generate optimal solutions of Problem 2.4 by the single generating function method. This is presented in the following theorem.

Theorem 2.9 ([50])**.** *The optimal input of Problem 2.4 is given as the state feedforward control*

$$u_k^* = -R^{-1} B^{\mathsf{T}} \left(\mathscr{U}_{2\text{b},k+1} x_{k+1} + \mathscr{V}_{2\text{b},k+1}^{\mathsf{T}} \lambda_N \right), \quad k = 0, 1, \cdots, N-1 \tag{2.139}$$

where

$$\lambda_N = \mathscr{W}_{2\text{b},0}^{-1} \left(x_N - \mathscr{V}_{2\text{b},0} x_0 \right). \tag{2.140}$$

2.3 Summary

This chapter introduces preliminaries of the Hamiltonian system and the generating functions. For both the continuous and discrete time optimal control problems, we give necessary and sufficient (only for continuous-time case) conditions for optimality, derive Hamilton–Jacobi equations and generating functions, provide optimal solutions (only for continuous-time case), present relations between the generating function and the value function, and exhibit the LQ cases. The latter Chapters 3, 4, and 5 are all developed based on these preliminaries.

Chapter 3

Continuous-time state constrained LQ optimal control problem

The generating function method is effective in solving the Hard Constrained Problem as introduced in Chapter 2. In Hard Constrained Problem, there only has the terminal state constraint, but does not contain any general inequality constraints, especially the inequality state constraints. So far, none of the existing literature has studied the optimal control problem with inequality constraints by using the generating function methods.

Our goal of this chapter is to extend the generating function method to the inequality state constrained problem. However, the generating function is one of the indirect methods that are difficult in handling inequality constraints in contrast with the direct methods. The idea here is to convert the constrained problem to an unconstrained problem such that we can avoid dealing with constraints. We employ the penalty function to achieve the goals. There exist several related papers [57, 58, 59]. However, all of them are for the Soft Constrained Problem that can not be readily extended to the generating function method for the Hard Constrained Problem.

This chapter extends the generating function method to the Hard Constrained Problem with inequality state constraints, i.e. the classical path and terminal state constrained LQ optimal control problem [10]. First in Section 3.1, we formulate and design the constrained problem and the penalized Hard Constrained Problem respectively, show their convex properties, and further exhibit the convergence of the minimum cost function value and optimal solutions between these two problems under a mild condition. Second in Section 3.2, due to the technique of the Taylor series expansion, the partial differential Hamilton–Jacobi equation is reduced to the ordinary differential equations for the generating function coefficients. We give the recursive condition to eliminate the coupling relations between the coefficients with lower and higher indices in these ordinary differential equations so that they can be solved recursively. This guarantees the penalized Hard Constrained Problem to be successfully solved by generating function method. Based on this, in Section 3.3, we summarize how to design penalties which is suit for the generating function method, and gives an algorithm presents how to generate optimal solutions repetitively for different boundary conditions. At last in Section 3.4, we give two examples to illustrate the effectiveness of the developed method. Section 3.5 summarizes this chapter.

3.1 Problem conversion

In this section, we formulate and design the constrained and penalized problems, show their convexities in Section 3.1.1 and 3.1.2, respectively. Based on this, we exhibit in Section 3.1.3 the convergence of the minimum value and optimal solutions as the penalty factor goes to zero. In light of this, we can select a rather small factor such that we convert the constrained problem to the penalized problem, which is possible to be solved by the generating function method.

3.1.1 Constrained problem and its convexity

Consider the following continuous-time state constrained LQ optimal control problem.

Problem 3.1.

$$\min_{u} \int_{t_0}^{t_f} \frac{1}{2}\left(x(t)^{\mathsf{T}} Q x(t) + u(t)^{\mathsf{T}} R u(t)\right) \mathrm{d}t \tag{3.1}$$

$$\text{s.t. } \dot{x}(t) = Ax(t) + Bu(t), \quad t \in [t_0, t_f] \tag{3.2}$$

$$x(t_0) = x_{\text{init}}, \quad x(t_f) = x_{\text{term}} \tag{3.3}$$

$$C_k\left(x(t)\right) \leqslant 0, \quad k = 1, 2, \cdots, s, \quad \forall t \in [t_0, t_f] \tag{3.4}$$

where the state path constraint is defined as $C_k \colon \mathbb{R}^n \to \mathbb{R}$, $k = 1, 2, \cdots, s$.

For the state path constraint, we make the following assumption.

Assumption 3.1. *Assume that*

(i) *C_k is a convex function, $\forall k = 1, 2, \cdots, s$;*

(ii) *$C_k(x_{\text{init}}) < 0$ and $C_k(x_{\text{term}}) < 0$, $\forall k = 1, 2, \cdots, s$.*

Notice the problem (3.1)–(3.3), it is a standard Hard Constraint Problem [36], since the terminal boundary condition is prescribed to a fixed point. Together with the inequality state constraints in (3.4) along the time interval $t \in [t_0, t_f]$, we form a path and terminal state constrained LQ optimal control problem. The generating function method will be developed to solve such Problem 3.1 later in the chapter.

Now to study Problem 3.1, we give the following assumption which will be used to prove Theorem 3.3 in the next subsection.

Assumption 3.2. *Assume that the optimal input of Problem 3.1, i.e. $u^*(t)$, is continuous in t.*

It is easy to know that for each input u, the dynamics (3.2) satisfying the initial boundary condition $x(t_0) = x_{\text{init}}$ in (3.3), i.e. the initial value problem, has a unique solution x^u. Based on this, we then give the following definition that will be used throughout this chapter.

Definition 3.1. Define three sets

(i) $\mathcal{U}^{\text{f}} := \{u \in L^\infty([t_0, t_f], \mathbb{R}^m) \mid x^u(t_f) = x_{\text{term}}\}$

(ii) $\mathcal{U}^{\text{p}} := \{u \in L^\infty([t_0, t_f], \mathbb{R}^m) \mid C_k(x^u(t)) \leqslant 0, k = 1, 2, \cdots, s, \forall t \in [t_0, t_f]\}$

(iii) $\mathcal{U}^{\mathrm{p}0} := \{u \in L^\infty([t_0, t_\mathrm{f}], \mathbb{R}^m) \mid C_k(x^u(t)) < 0, k = 1, 2, \cdots, s, \forall t \in [t_0, t_\mathrm{f}]\}$

For these sets, we have the following assumption.

Assumption 3.3. *Assume that the sets $\mathcal{U}^\mathrm{f}$, $\mathcal{U}^{\mathrm{p}0}$, and their intersection $\mathcal{U}^\mathrm{f} \cap \mathcal{U}^{\mathrm{p}0}$ are all nonempty.*

Remark 3.1. Based on Assumption 3.3, it is straightforward to know that the set $\mathcal{U}^\mathrm{p}$ and another intersection $\mathcal{U}^\mathrm{f} \cap \mathcal{U}^\mathrm{p}$ are also nonempty.

Next, we will show the convexity of Problem 3.1. To this end, we first reformulate the original problem as the following Problem 3.1′.

Problem 3.1′.

$$\min_{u \in \mathcal{U}^\mathrm{f} \cap \mathcal{U}^\mathrm{p}} \left(J(u) := \int_{t_0}^{t_\mathrm{f}} \frac{1}{2} \left((x^u(t))^\mathsf{T} Q (x^u(t)) + u(t)^\mathsf{T} R u(t) \right) \mathrm{d}t \right)$$

Then, it is easy for us to show its convex properties, including convexities of the set $\mathcal{U}^\mathrm{f} \cap \mathcal{U}^\mathrm{p}$ and the cost function J.

Proposition 3.1. *Under Assumptions 3.1 and 3.3, the set $\mathcal{U}^\mathrm{f} \cap \mathcal{U}^\mathrm{p}$ of Problem 3.1′ is a convex set of the input u .*

Proof. To prove this theorem, we give two inputs u^1 and $u^2 \in \mathcal{U}^\mathrm{f} \cap \mathcal{U}^\mathrm{p}$, and the parameter $0 < \theta < 1$. Due to the linear dynamics (3.2), we have

$$x^{\theta u^1 + (1-\theta)u^2}(t_\mathrm{f}) = \theta x^{u^1}(t_\mathrm{f}) + (1-\theta) x^{u^2}(t_\mathrm{f}) = x(t_\mathrm{f}) = x_\mathrm{term}. \tag{3.5}$$

Further, since the constraint function C_k is convex due to Assumption 3.1, we have

$$C_k\left(x^{\theta u^1 + (1-\theta)u^2}\right) = C_k\left(\theta x^{u^1} + (1-\theta) x^{u^2}\right) \leqslant \theta C_k\left(x^{u^1}\right) + (1-\theta) C_k\left(x^{u^2}\right) \leqslant 0. \tag{3.6}$$

In summary, (3.5) and (3.6) imply $\theta u^1 + (1-\theta)u^2 \in \mathcal{U}^\mathrm{f} \cap \mathcal{U}^\mathrm{p}$ such that the set $\mathcal{U}^\mathrm{f} \cap \mathcal{U}^\mathrm{p}$ of the input u is a convex set. □

Proposition 3.2. *Under Assumption 3.3, the cost function $J(u)$ of Problem 3.1′ is a strongly convex function in u, $\forall u \in \mathcal{U}^\mathrm{f} \cap \mathcal{U}^\mathrm{p}$, i.e. it satisfies*

$$J(\theta u^1 + (1-\theta)u^2) \leqslant \theta J(u^1) + (1-\theta) J(u^2) - \frac{1}{2} r \theta (1-\theta) \left\| u^1 - u^2 \right\|_{L^2}^2, \quad \forall u^1, u^2 \in \mathcal{U}^\mathrm{f} \cap \mathcal{U}^\mathrm{p} \tag{3.7}$$

for some $r > 0$ and $0 < \theta < 1$.

Proof. For the two inputs u^1 and $u^2 \in \mathcal{U}^\mathrm{f} \cap \mathcal{U}^\mathrm{p}$, and the parameter $0 < \theta < 1$, we have inequalities

$$\begin{aligned} &\theta J(u^1) + (1-\theta) J(u^2) - J(\theta u^1 + (1-\theta)u^2) \\ =&\frac{1}{2} \int_{t_0}^{t_\mathrm{f}} \theta(1-\theta) \left((x^{u^1} - x^{u^2})^\mathsf{T} Q (x^{u^1} - x^{u^2}) + (u^1 - u^2)^\mathsf{T} R (u^1 - u^2) \right) \mathrm{d}t \end{aligned}$$

$$
\begin{aligned}
&\geqslant \frac{1}{2}\int_{t_0}^{t_\mathrm{f}} \theta(1-\theta)(u^1-u^2)^\mathsf{T} R(u^1-u^2)\mathrm{d}t \\
&\geqslant \frac{1}{2} r\theta(1-\theta)\int_{t_0}^{t_\mathrm{f}} \left\|u^1-u^2\right\|^2 \mathrm{d}t \\
&= \frac{1}{2} r\theta(1-\theta)\left\|u^1-u^2\right\|_{L^2}^2
\end{aligned}
$$

where the positive r can be assigned as $r \leqslant \sigma_{\min}(R)$. By summarizing the inequalities, we have (3.7) so that the cost function $J(u)$ is a strongly convex function in u, $\forall u \in \mathcal{U}^\mathrm{f} \cap \mathcal{U}^\mathrm{p}$. □

This strongly convex property will be used to prove Theorem 3.2 about the convergence in the next subsection.

Remark 3.2. Since strong convexity is stronger than strict convexity, Proposition 3.2 also implies that the cost function $J(u)$ is a strictly convex function in u, $\forall u \in \mathcal{U}^\mathrm{f} \cap \mathcal{U}^\mathrm{p}$. Then, by summarizing Propositions 3.1 and 3.2, we know that Problem 3.1′ is the problem minimizing a strictly convex cost function of u over a nonempty convex set $\mathcal{U}^\mathrm{f} \cap \mathcal{U}^\mathrm{p}$ in u space. Hence Problem 3.1′ (Problem 3.1) has unique global minimizer u^* [17].

3.1.2 Penalized problem and its convexity

We design the following penalized Hard Constraint Problem by adding a penalty term in the cost function.

Problem 3.2.

$$
\begin{aligned}
&\min_u \int_{t_0}^{t_\mathrm{f}} \left(\frac{1}{2}\left(x(t)^\mathsf{T} Q x(t) + u(t)^\mathsf{T} R u(t)\right) + \mu P\left(x(t)\right)\right)\mathrm{d}t \\
&\text{s.t. } \dot{x}(t) = Ax(t) + Bu(t), \quad t \in [t_0, t_\mathrm{f}] \\
&\qquad x(t_0) = x_\mathrm{init}, \quad x(t_\mathrm{f}) = x_\mathrm{term}
\end{aligned}
\tag{3.8}
$$

where μ is a positive factor which penalizes the closeness to the constraint boundaries, and the penalty function $P(x) \equiv P'(C(x))$ where $C(x) = [C_1(x), C_2(x), \cdots, C_s(x)]^\mathsf{T}$ and $P' \colon \mathbb{R}^s \to \mathbb{R}$. For this general penalty function, we need the following assumption.

Assumption 3.4. *Assume that*

(i) $P(x)$ *is a convex function of* $x \in \mathcal{X}^{\mathrm{p}0}$

(ii) $P(x) \geqslant 0$, $\forall x \in \mathcal{X}^{\mathrm{p}0}$

(iii) $P(x) \to +\infty$ *when* x *approaches the boundary of* $\mathcal{X}^{\mathrm{p}0}$ *from its interior*

where $\mathcal{X}^{\mathrm{p}0} := \{x \in L^\infty([t_0, t_\mathrm{f}], \mathbb{R}^n) \mid C_k(x) < 0, k = 1, 2, \cdots, s\}$ *is the set of the state satisfying the strict path constraint.*

Remark 3.3. In Problem 3.2 with starting and ending points (x_{init} and x_{term}) in the interior, the value of the penalty grows sharply when x (driven by u) approaches the boundary of the path constraint such that it can prevent the state trajectory violating the constraints. This implies $u \in \mathcal{U}^{\text{f}} \cap \mathcal{U}^{\text{p0}}$ in Problem 3.2 in fact [58], which is the goal achieved by adding the penalty. Correspondingly, the augmented cost function in (3.8) then can be minimized in the absence of the path constraint, yielding a biased estimate of the solution of Problem 3.1. It is natural to imagine that we can set the factor μ small enough to reduce the bias such that there may exist the convergence. This will be discussed in the next subsection.

Now, to show the convexity of Problem 3.2, we reformulate it as well in the following.

Problem 3.2′.

$$\min_{u\in\mathcal{U}^{\text{f}}} \left(J_p(u,\mu) \colon = \int_{t_0}^{t_{\text{f}}} \left(\frac{1}{2} \left((x^u(t))^{\mathsf{T}} Q\,(x^u(t)) + u(t)^{\mathsf{T}} R u(t) \right) + \mu P\,(x^u(t)) \right) \mathrm{d}t \right). \quad (3.9)$$

Then, we exhibit the convex properties in the following theorem, including convexities of the set $\mathcal{U}^{\text{f}}$ and the cost function J_p.

Proposition 3.3. *Under Assumption 3.3, the set $\mathcal{U}^{\text{f}}$ of Problem 3.2′ is a convex set of the input u.*

The proof here is straightforward by (3.5).

Proposition 3.4. *Under Assumptions 3.1, 3.3, and 3.4, the penalized cost function $J_p(u,\mu)$ of Problem 3.2′ is strictly convex in u, $\forall u \in \mathcal{U}^{\text{f}}$ and $\forall \mu > 0$.*

Proof. Since $Q \succcurlyeq 0$ and $R \succ 0$, the first two terms $\frac{1}{2}(x^u)^{\mathsf{T}} Q x^u$ and $\frac{1}{2} u^{\mathsf{T}} R u$ in the integrand of (3.9) can be treated as convex and strictly convex functions respectively. Further, under Assumption 3.4(i) and (ii) for the penalty with the positive factor μ, it is clear for us to know that $J_p(u,\mu)$ is strictly convex in u, $\forall u \in \mathcal{U}^{\text{f}}$ and $\forall \mu > 0$. □

Remark 3.4. Propositions 3.3 and 3.4 show that Problem 3.2′ is the problem minimizing a strictly convex cost function of u over a nonempty convex set $\mathcal{U}^{\text{f}}$ in u space. Hence Problem 3.2′ (Problem 3.2) has unique global minimizer $u_p^*(\mu)$ for each specified μ.

3.1.3 Convergence

We will exhibit the convergent properties of minimum cost function value and the optimal solutions in this subsection. Before this, notice the definition of J_p in (3.9), we can rewrite it as

$$J_p(u,\mu) = J(u) + \int_{t_0}^{t_{\text{f}}} \mu P(x^u)\mathrm{d}t \quad (3.10)$$

for the sake of clarity. Now first, we denote J^* as the minimum cost function value of Problem 3.1, and present the following theorem as a preparation.

Theorem 3.1. *Under Assumptions 3.1, 3.3, and 3.4, for the penalized cost function of Problem 3.2, we have the following convergent properties*

(i) $\lim_{\mu\to 0} J(u_p^*(\mu)) = J^*$;

(ii) $\lim_{\mu\to 0} \int_{t_0}^{t_\mathrm{f}} \mu P(x^{u_p^*(\mu)})\mathrm{d}t = 0.$

For the proof see [57, 58, 64], since these proofs also work for the case of Problem 3.2. Theorem 3.1 presents the convergences of the two summands in the right hand side of (3.10).

Second, we will exhibit the three main convergences. By combining Theorem 3.1(i) and (ii), we have the following corollary readily.

Corollary 3.1. *Under Assumptions 3.1, 3.3, and 3.4, for the minimum cost function values of Problems 3.1′ and 3.2′, we have the convergence*

$$\lim_{\mu\to 0} J_p\left(u_p^*(\mu), \mu\right) = J^*.$$

Theorem 3.2. *Under Assumptions 3.1, 3.3, and 3.4, for the optimal inputs of Problems 3.1′ and 3.2′, we have the convergence*

$$\lim_{\mu\to 0} \|u_p^*(\mu) - u^*\|_{L^2} = 0. \tag{3.11}$$

This theorem can be proven mainly based on Proposition 3.2, which is an assumption in [58]. Under such assumption, [58] proves the convergence of the input. Since the logic is similar, we present the proof here in brief to make the theorem self-contained.

Proof. Letting $\theta = \frac{1}{2}$, $u^1 = u^* \in \mathcal{U}^\mathrm{f} \cap \mathcal{U}^\mathrm{p}$, and $u^2 = u_p^*(\mu) \in \mathcal{U}^\mathrm{f} \cap \mathcal{U}^\mathrm{p}$ in Proposition 3.2, then (3.7) reads

$$\frac{r}{8}\left\|u^* - u_p^*(\mu)\right\|_{L^2}^2 \leqslant \frac{1}{2}J(u^*) + \frac{1}{2}J(u_p^*(\mu)) - J\left(\frac{u^* + u_p^*(\mu)}{2}\right) \tag{3.12}$$

where $\frac{u^*+u_p^*(\mu)}{2} \in \mathcal{U}^\mathrm{f} \cap \mathcal{U}^\mathrm{p}$ as well. Further, we know

$$J(u^*) \leqslant J\left(\frac{u^* + u_p^*(\mu)}{2}\right). \tag{3.13}$$

Then by the substitution of (3.13), (3.12) leads to

$$\frac{r}{8}\left\|u^* - u_p^*(\mu)\right\|_{L^2}^2 \leqslant \frac{1}{2}J(u_p^*(\mu)) - \frac{1}{2}J(u^*).$$

Now, by using Theorem 3.1(i) (note that $J^* = J(u^*)$), we prove (3.11). □

Theorem 3.3. *Under Assumptions 3.1–3.4, for the optimal states of Problems 3.1′ and 3.2′, we have the convergence*

$$\lim_{\mu\to 0} \|x^{u_p^*(\mu)} - x^{u^*}\|_{L^\infty} = 0. \tag{3.14}$$

Proof. We know the explicit expression of x^u is

$$x(t) = e^{A(t-t_0)}x_0 + \int_{t_0}^{t} e^{A(t-\tau)}Bu(\tau)\mathrm{d}\tau$$

then to prove (3.14) amounts to proving

$$\sup_{t_0 \leqslant t \leqslant t_\mathrm{f}} \left\| \int_{t_0}^{t} e^{A(t-\tau)}B\left(u^*(\tau) - u_p^*(\mu,\tau)\right)\mathrm{d}\tau \right\| \tag{3.15}$$

going to zero as μ going to zero. This is definitely the case, because the term can be estimated as (the following first inequality holds due to the continuity of u^* and u_p^* by Assumption 3.2 and Theorem 2.1, respectively)

$$\begin{aligned}
&\left\| \int_{t_0}^{t} e^{A(t-\tau)}B\left(u^*(\tau) - u_p^*(\mu,\tau)\right)\mathrm{d}\tau \right\| \\
\leqslant &\int_{t_0}^{t} \left\| e^{A(t-\tau)}B\left(u^*(\tau) - u_p^*(\mu,\tau)\right)\right\| \mathrm{d}\tau \\
\leqslant &K\|B\| \int_{t_0}^{t} \left\| u^*(\tau) - u_p^*(\mu,\tau)\right\| \mathrm{d}\tau
\end{aligned}$$

where K is the upper bound of the semigroup generated by A (any semigroup has an upper bound on time intervals of finite length), such that

$$(3.15) \leqslant K\|B\| \left\| u^* - u_p^*(\mu)\right\|_{L^1}.$$

Since $L^2([t_0, t_\mathrm{f}])$ norm is stronger than $L^1([t_0, t_\mathrm{f}])$ norm, we can deduce $\|u^* - u_p^*(\mu)\|_{L^1} \to 0$ from $\|u^* - u_p^*(\mu)\|_{L^2} \to 0$ (Theorem 3.2). Therefore, as $\mu \to 0$, (3.15) goes to zero. This proves (3.14). □

Note that the proof of Theorem 3.3 here is more clear and targeted than the one in [58].

In summary, by the convexity and convergence analysis in this section, we know that both Problem 3.1 and 3.2 are strictly convex problems such that they have unique global minimizers, moreover the minimum cost function value as well as the optimal input and state of Problem 3.2 converge to the ones of Problem 3.1 as the factor $\mu \to 0$. Hence we can select a rather small factor μ to form a penalized problem approximating the constrained problem. From this viewpoint, we convert the constrained Problem 3.1 to the penalized Problem 3.2. Such problem conversion enables us to solve the original problem by the generating function method indirectly.

3.2 Generating function method

In this section, based on the introduction of Taylor series solution to Hamilton–Jacobi equation in Section 3.2.1, we develop a recursive condition that guarantees the designed Problem 3.2 can be successfully solved via generating functions in Section 3.2.2.

3.2.1 Taylor series solution to Hamilton–Jacobi equation

Theorem 2.3 gives the optimal input of Problem 3.2 analytically as the state feedback control via the generating function. If we can solve the Hamilton–Jacobi equation for the generating function, it is easy for us to generate the optimal input by (2.42). Since the Hamilton–Jacobi equation (2.33) is a nonlinear partial differential equation, it is difficult to find its analytic solution so that we need the numerical implementations to find its approximate solution. As mentioned in Section 2.1.3, Taylor series expansion is the most popular numerical method utilized for such a purpose [36, 43, 35], so here we will also use this technique.

First, we expand the nonlinear functions in the Hamilton–Jacobi equation (2.33), i.e. the generating function[†] and the penalty function, as Taylor series in their arguments about the origin. To do so, the following assumption is needed.

Assumption 3.5. *Assume that for Problem 3.2*

(i) $F_{2\mathrm{b}}(x, \lambda(t_{\mathrm{f}}), t)$ *is an analytic function of* x *and* $\lambda(t_{\mathrm{f}})$ *in their neighborhoods of the origin in* $\mathbb{R}^{2n}$*;*

(ii) $P(x)$ *is an analytic function of* x *in its neighborhood of the origin in* $\mathbb{R}^{n}$*;*

(iii) *The set* $\mathcal{X}^{\mathrm{p0}}$ *is equal to or a subset of a neighborhood of the origin for the state* x*.*

Based on this, we expand both $F_{2\mathrm{b}}(x, \lambda(t_{\mathrm{f}}), t)$ and $P(x)$ as Taylor series in their arguments about the origin up to a fixed order $\mathcal{N}$ as[‡]

$$\sum_{i=0}^{\mathcal{N}} \sum_{j=0}^{i} \left(\mathscr{F}_{(i,j)}(t) \cdot \left(x^{\otimes(i-j)} \otimes \lambda(t_{\mathrm{f}})^{\otimes j} \right) \right) \tag{3.16}$$

$$\sum_{i=0}^{\mathcal{N}} \left(\mathscr{P}_{(i)} \cdot x^{\otimes i} \right) \tag{3.17}$$

where $\mathscr{F}_{(i,j)}(t) \cdot \left(x^{\otimes(i-j)} \otimes \lambda(t_{\mathrm{f}})^{\otimes j} \right) \equiv \mathcal{F}_{(i,j)}(x, \lambda(t_{\mathrm{f}}), t)$ and $\mathscr{P}_{(i)} \cdot x^{\otimes i} \equiv \mathcal{P}_{(i)}(x)$. Here, $\mathscr{F}_{(i,j)}(t)$ is the coefficient of the (i, j)-th Taylor series term $\mathcal{F}_{(i,j)}(x, \lambda(t_{\mathrm{f}}), t)$ of the function $F(x, \lambda(t_{\mathrm{f}}), t)$, and $\mathscr{P}_{(i)}$ is the coefficient of the i-th Taylor series term $\mathcal{P}_{(i)}(x)$ of the function $P(x)$. Here, $\otimes$ is the Kronecker product [65, 66]. For example, if Y is an $m \times n$ matrix and Z is a $p \times q$ matrix, then the Kronecker product $Y \otimes Z$ is the $mp \times nq$ block matrix

$$Y \otimes Z = \begin{bmatrix} y_{11}Z & \cdots & y_{1n}Z \\ \vdots & \ddots & \vdots \\ y_{m1}Z & \cdots & y_{mn}Z \end{bmatrix}$$

where y_{ij} denotes the (i, j)-th element of the matrix Y. Moreover

$$Y^{\otimes i} = \underbrace{Y \otimes Y \otimes \cdots \otimes Y}_{i}.$$

[†] According to Theorem 2.3, we will use the generating function $F_{2\mathrm{b}}$ in this chapter.

[‡] In this chapter, only the generating function $F_{2\mathrm{b}}$ is used, so we do not add the subscript $2\mathrm{b}$ in its Taylor series terms and coefficients for convenience.

In (3.16) and (3.17), note that the generating function coefficients $\mathscr{F}_{(i,j)}(t)$'s are undetermined, while the penalty function coefficients $\mathscr{P}_{(i)}$'s are known. The objective here is to determine the unknown $\mathscr{F}_{(i,j)}(t)$'s.

By substituting the Taylor series of $F_{2\mathrm{b}}$ and P, i.e. (3.16) and (3.17), into the Hamiltonian[†] (with $\lambda = \frac{\partial F_{2\mathrm{b}}(x,\lambda(t_\mathrm{f}),t)}{\partial x}$ by (2.34))

$$H(x,\lambda) = \frac{1}{2}x^\mathsf{T} Q x + \lambda^\mathsf{T} A x - \frac{1}{2}\lambda^\mathsf{T} G \lambda + \mu P(x) \tag{3.18}$$

we get its power series form

$$\sum_{i=0}^{\mathcal{N}} \sum_{j=0}^{i} \left(\mathscr{H}_{(i,j)} \left(\mathscr{F}_{(\cdot,\cdot)}(t) \right) \cdot \left(x^{\otimes(i-j)} \otimes \lambda(t_\mathrm{f})^{\otimes j} \right) \equiv \mathcal{H}_{(i,j)} \left(x, \frac{\partial \mathcal{F}_{(\cdot,\cdot)}(x,\lambda(t_\mathrm{f}),t)}{\partial x} \right) \right)$$

where $\mathscr{H}_{(i,j)}$ is the coefficient of the (i,j)-th power series term $\mathcal{H}_{(i,j)}$ of the Hamiltonian. Based on this, by collecting the terms with the same variable $\left(x^{\otimes(i-j)} \otimes \lambda(t_\mathrm{f})^{\otimes j}\right)$ from the Hamilton–Jacobi equation (2.33), we get the expanded Hamilton–Jacobi equation

$$\sum_{i=0}^{\mathcal{N}} \sum_{j=0}^{i} \left(\frac{\partial \mathcal{F}_{(i,j)}(x,\lambda(t_\mathrm{f}),t)}{\partial t} + \mathcal{H}_{(i,j)} \left(x, \frac{\partial \mathcal{F}_{(\cdot,\cdot)}(x,\lambda(t_\mathrm{f}),t)}{\partial x} \right) \right) = 0 \tag{3.19}$$

Based on the above expansions, now we present the following theorem to determine the generating function coefficients $\mathscr{F}_{(i,j)}(t)$'s.

Theorem 3.4 ([43]). *Under Assumptions 3.1 and 3.3–3.5, for Problem 3.2, the coefficients $\mathscr{F}_{(i,j)}(t)$'s of the generating function $F_{2\mathrm{b}}$ are determined by solving the following ordinary differential equations ($t \in [t_0, t_\mathrm{f}]$)*

$$\dot{\mathscr{F}}_{(i,j)}(t) = -\mathscr{H}_{(i,j)} \left(\mathscr{F}_{(\cdot,\cdot)}(t) \right), \quad j = 0, 1, \cdots, i \text{ and } i = 0, 1, \cdots, \mathcal{N} \tag{3.20}$$

with their terminal conditions

$$\mathscr{F}_{(i,j)}(t_\mathrm{f}) = \begin{cases} I, & i = 2, j = 1 \\ 0, & \text{other cases} \end{cases} \tag{3.21}$$

where $I \in \mathbb{R}^{1\times nn}$ with all its elements equal to one.

By using Taylor series numerical techniques, the partial differential Hamilton–Jacobi equation (2.33) is reduced to ordinary differential equations (3.20). Once we obtain the generating function coefficients $\mathscr{F}_{(i,j)}$'s, we obtain the generating function. Finally by substituting it into Theorem 2.3, we get the optimal input. This is the whole procedure how we generate optimal solutions by the generating function method.

[†]In fact, the Hamiltonian should be expressed as $H(x,\lambda,\mu)$. However, since μ is treated as a parameter (not a variable) in this section, we write the Hamiltonian as $H(x,\lambda)$ for convenience.

3.2.2 Recursive condition

Notice the ordinary differential equations (3.20), in its right hand side there exist coefficients $\mathscr{F}_{(\cdot,\cdot)}(t)$ whose index are greater than (i,j), e.g. $\mathscr{F}_{(i+1,j)}(t)$. Due to this, we can not solve $\mathscr{F}_{(i,j)}(t)$'s recursively from $(i,j)=(0,0)$ to the truncated order $(\mathcal{N},\mathcal{N})$. However, the recursiveness can be achieved by adding a mild condition. This is presented in the following theorem.

Theorem 3.5. *Under Assumptions 3.1 and 3.3–3.5, for Problem 3.2, if the penalty function coefficient $\mathscr{P}_{(1)}=0$, the ordinary differential equations (3.20) can be solved recursively for the generating function coefficients $\mathscr{F}_{(i,j)}(t)$ with respect to the Taylor series order index (i,j).*

Proof. In principle, to prove Theorem 3.5, we should concentrate on the ordinary differential equation (3.20). But here we focus on (3.19) instead, for the reason that it can present clearer coupling relations.

We first show the exact expression of (3.19) in the following where we will use $\mathcal{F}_{(i,j)}$ and $\mathcal{P}_{(i)}$ short for $\mathcal{F}_{(i,j)}(x,\lambda(t_{\mathrm{f}}),t)$ and $\mathcal{P}_{(i)}(x)$, respectively

$$\frac{\partial \mathcal{F}_{(0,0)}}{\partial t} = -\mu\mathcal{P}_{(0)} \tag{3.19a}$$

$$\frac{\partial \mathcal{F}_{(1,0)}}{\partial t} = \frac{1}{2}\left(\frac{\partial \mathcal{F}_{(1,0)}}{\partial x}\right)^{\mathsf{T}} G\left(\frac{\partial \mathcal{F}_{(2,0)}}{\partial x}\right) + \frac{1}{2}\left(\frac{\partial \mathcal{F}_{(2,0)}}{\partial x}\right)^{\mathsf{T}} G\left(\frac{\partial \mathcal{F}_{(1,0)}}{\partial x}\right) - \left(\frac{\partial \mathcal{F}_{(1,0)}}{\partial x}\right)^{\mathsf{T}} Ax - \mu\mathcal{P}_{(1)} \tag{3.19b}$$

$$\frac{\partial \mathcal{F}_{(1,1)}}{\partial t} = \frac{1}{2}\left(\frac{\partial \mathcal{F}_{(1,0)}}{\partial x}\right)^{\mathsf{T}} G\left(\frac{\partial \mathcal{F}_{(2,1)}}{\partial x}\right) + \frac{1}{2}\left(\frac{\partial \mathcal{F}_{(2,1)}}{\partial x}\right)^{\mathsf{T}} G\left(\frac{\partial \mathcal{F}_{(1,0)}}{\partial x}\right) \tag{3.19c}$$

$$\begin{aligned}\frac{\partial \mathcal{F}_{(2,0)}}{\partial t} =& \frac{1}{2}\left(\frac{\partial \mathcal{F}_{(1,0)}}{\partial x}\right)^{\mathsf{T}} G\left(\frac{\partial \mathcal{F}_{(3,0)}}{\partial x}\right) + \left(\frac{\partial \mathcal{F}_{(2,0)}}{\partial x}\right)^{\mathsf{T}} G\left(\frac{\partial \mathcal{F}_{(2,0)}}{\partial x}\right) + \frac{1}{2}\left(\frac{\partial \mathcal{F}_{(3,0)}}{\partial x}\right)^{\mathsf{T}} G\left(\frac{\partial \mathcal{F}_{(1,0)}}{\partial x}\right) \\ & - \left(\frac{\partial \mathcal{F}_{(2,0)}}{\partial x}\right)^{\mathsf{T}} Ax - \frac{1}{2}x^{\mathsf{T}}Qx - \mu\mathcal{P}_{(2)}\end{aligned} \tag{3.19d}$$

$$\vdots$$

$$\frac{\partial \mathcal{F}_{(i,j)}}{\partial t} = \begin{cases} \frac{1}{2}\sum_{i_1=1}^{i+1}\left(\frac{\partial \mathcal{F}_{(i,0)}}{\partial x}\right)^{\mathsf{T}} G\left(\frac{\partial \mathcal{F}_{(i+2-i_1,0)}}{\partial x}\right) - \left(\frac{\partial \mathcal{F}_{(i,0)}}{\partial x}\right)^{\mathsf{T}} Ax - \mu\mathcal{P}_{(i)}, & j=0 \\ \frac{1}{2}\sum_{i_1=1}^{i+1}\sum_{j_1=\max\{0,j+i_1-i-2\}}^{\min\{i_1,j\}}\left(\frac{\partial \mathcal{F}_{(i_1,j_1)}}{\partial x}\right)^{\mathsf{T}} G\left(\frac{\partial \mathcal{F}_{(i+2-i_1,j-j_1)}}{\partial x}\right) - \left(\frac{\partial \mathcal{F}_{(i,j)}}{\partial x}\right)^{\mathsf{T}} Ax, & \text{other cases} \\ \frac{1}{2}\sum_{i_1=1}^{i+1}\left(\frac{\partial \mathcal{F}_{(i_1,i_1-1)}}{\partial x}\right)^{\mathsf{T}} G\left(\frac{\partial \mathcal{F}_{(i+2-i_1,j-i_1+1)}}{\partial x}\right), & j=i \end{cases} \tag{3.19e}$$

$$\vdots$$

To clearly show the structure of (3.19e), we expand the three formulae in its right hand side as

$$\begin{cases} \frac{1}{2}\left(\frac{\partial \mathcal{F}_{(1,0)}}{\partial x}\right)^{\mathsf{T}} G\left(\frac{\partial \mathcal{F}_{(i+1,0)}}{\partial x}\right) + \frac{1}{2}\left(\frac{\partial \mathcal{F}_{(2,0)}}{\partial x}\right)^{\mathsf{T}} G\left(\frac{\partial \mathcal{F}_{(i,0)}}{\partial x}\right) + \cdots \\ \quad + \frac{1}{2}\left(\frac{\partial \mathcal{F}_{(i,0)}}{\partial x}\right)^{\mathsf{T}} G\left(\frac{\partial \mathcal{F}_{(2,0)}}{\partial x}\right) + \frac{1}{2}\left(\frac{\partial \mathcal{F}_{(i+1,0)}}{\partial x}\right)^{\mathsf{T}} G\left(\frac{\partial \mathcal{F}_{(1,0)}}{\partial x}\right) - \left(\frac{\partial \mathcal{F}_{(i,0)}}{\partial x}\right)^{\mathsf{T}} Ax - \mu\mathcal{P}_{(i)}, & j=0 \\ \frac{1}{2}\left(\frac{\partial \mathcal{F}_{(1,0)}}{\partial x}\right)^{\mathsf{T}} G\left(\frac{\partial \mathcal{F}_{(i+1,j)}}{\partial x}\right) + \frac{1}{2}\left(\frac{\partial \mathcal{F}_{(2,0)}}{\partial x}\right)^{\mathsf{T}} G\left(\frac{\partial \mathcal{F}_{(i,j)}}{\partial x}\right) + \cdots \\ \quad + \frac{1}{2}\left(\frac{\partial \mathcal{F}_{(i,j)}}{\partial x}\right)^{\mathsf{T}} G\left(\frac{\partial \mathcal{F}_{(2,0)}}{\partial x}\right) + \frac{1}{2}\left(\frac{\partial \mathcal{F}_{(i+1,j)}}{\partial x}\right)^{\mathsf{T}} G\left(\frac{\partial \mathcal{F}_{(1,0)}}{\partial x}\right) - \left(\frac{\partial \mathcal{F}_{(i,j)}}{\partial x}\right)^{\mathsf{T}} Ax, & \text{other cases} \\ \frac{1}{2}\left(\frac{\partial \mathcal{F}_{(1,0)}}{\partial x}\right)^{\mathsf{T}} G\left(\frac{\partial \mathcal{F}_{(i+1,j)}}{\partial x}\right) + \frac{1}{2}\left(\frac{\partial \mathcal{F}_{(2,1)}}{\partial x}\right)^{\mathsf{T}} G\left(\frac{\partial \mathcal{F}_{(i,j-1)}}{\partial x}\right) + \cdots \\ \quad + \frac{1}{2}\left(\frac{\partial \mathcal{F}_{(i,j-1)}}{\partial x}\right)^{\mathsf{T}} G\left(\frac{\partial \mathcal{F}_{(2,1)}}{\partial x}\right) + \frac{1}{2}\left(\frac{\partial \mathcal{F}_{(i+1,j)}}{\partial x}\right)^{\mathsf{T}} G\left(\frac{\partial \mathcal{F}_{(1,0)}}{\partial x}\right), & j=i \end{cases}$$

From the above expressions, now it is easy to find that the higher index coefficient, i.e. $\mathcal{F}_{(i+1,j)}$, exists in the (i,j)-th equation (3.19e). This means $\mathcal{F}_{(i,j)}$ is coupled with $\mathcal{F}_{(i+1,j)}$, i.e. $\dot{\mathscr{F}}_{(i,j)}(t)$ is coupled with $\mathscr{F}_{(i+1,j)}(t)$. Such phenomenon occurs in all the equations from (3.19b)–(3.19e) and so on. Hence after we reduce (3.19) to the ordinary differential equations (3.20), we can not solve them for the generating function coefficients recursively from $(i,j)=(0,0)$ to the truncated order $(\mathcal{N},\mathcal{N})$.

Further, it can be found that $\mathcal{F}_{(i+1,j)}$ is multiplied by $\mathcal{F}_{(1,0)}$, which is the only manner that $\mathcal{F}_{(i+1,j)}$ exists in (3.19e). This is also the same in the first three equations (3.19b)–(3.19d).

Based on the above, now notice the equation (3.19b), if $\mathcal{P}_{(1)}=0$ (so that $\mathscr{P}_{(1)}=0$) then each term in its right hand side has the factor $\mathcal{F}_{(1,0)}$. After we reduce (3.19b) to the ordinary differential equation

$$\dot{\mathscr{F}}_{(1,0)} = \mathscr{F}_{(1,0)} \cdot \Upsilon$$

where Υ is an expression. It is clear that $\mathscr{F}_{(1,0)}$ is locally Lipschitz so that the above ordinary differential equation for $\mathscr{F}_{(1,0)}$ has unique solution locally around the origin. Accordingly, it can be known that the solution is $\mathscr{F}_{(1,0)}=0$ (so that $\mathcal{F}_{(1,0)}=0$). Based on this, terms with both $\mathcal{F}_{(1,0)}$ and $\mathcal{F}_{(i+1,j)}$ are disappeared such that $\mathcal{F}_{(i,j)}$ will not be coupled with $\mathcal{F}_{(i+1,j)}$ again in (3.19), i.e. $\dot{\mathscr{F}}_{(i,j)}(t)$ will not be coupled with $\mathscr{F}_{(i+1,j)}(t)$ in (3.20). Therefore, the ordinary differential equations (3.20) can be solved recursively for the generating function coefficients $\mathscr{F}_{(i,j)}(t)$ with respect to the Taylor series order index (i,j). □

This theorem shows that if the penalty satisfies the presented condition $\mathscr{P}_{(1)}=0$, we can solve the ordinary differential equations (3.20) recursively for the generating function coefficients such that the designed penalized Problem 3.2 can be successfully solved by the generating function method.

3.3 Penalty design and generating function based algorithm

We exhibit how to design penalties for the generating function method in Section 3.3.1, and give an algorithm summarizing how to generate optimal solutions repetitively for different boundary conditions by generating functions in Section 3.3.2.

3.3.1 Penalty design

In our developed generating function method, the penalty plays the most important roles that how to design the penalty (penalized problem) suitable for the generating function is the key point. In this chapter, when we design the penalty function, we should pay attention to Assumption 3.4, Assumption 3.5(ii), and the recursive condition in Theorem 3.5. According to these conditions, we can design it by selecting the conventional penalty, e.g. inverse penalty or logarithmic penalty, and subtracting its first order Taylor series term with respect to the variable x. Specifically, we can design the penalty function as

$$P(x) = \sum_{k=1}^{s} \left(-\frac{1}{C_k(x)} + \left(\frac{\partial}{\partial x} \frac{1}{C_k(x)} \bigg|_{x=0} \right)^{\mathsf{T}} \cdot x \right) \tag{3.22}$$

based on the inverse penalty, or

$$P(x)=\sum_{k=1}^{s}\left(-\log\big(-C_k(x)\big)+\left(\frac{\partial}{\partial x}\log\big(-C_k(x)\big)\Big|_{x=0}\right)^{\mathsf{T}}\cdot x\right) \tag{3.23}$$

based on the logarithmic penalty.

For the convexity of the above designed penalties, since the second term in the bracket of (3.22) or (3.23) is linear, the convexity of the penalty function depends on the first term. Then according to the fact that the reciprocal of a real positive concave function is a convex function, and the logarithm of a real positive concave function is a concave function, it is clear that both (3.22) and (3.23) are convex functions.

3.3.2 Algorithm for different boundary conditions

It can be found from the above section that generating function (coefficients) are the same for different boundary conditions x_0 and x_{f}. In light of this, we can move the computation of these generating function (coefficients) to the off-line part, i.e. compute and save them in advance. Later, during the on-line calculation, we only need to read these generating function (coefficients) and substitute them to (2.42) to generate different optimal solutions for different boundary conditions. This method does not need to resolve the Hamilton–Jacobi equation repetitively for each different boundary conditions like the conventional dynamic programming. From this viewpoint, the method reduces the computational burden and is useful for on-line repetitive solutions generation for different boundary conditions. This is the computational advantage of the generating function method.

The detailed procedure about how to generate optimal state and input for different boundary conditions is summarized in the following two algorithms.

$\mu \leftarrow \mu_0$; /* set penalty factor */
$\mathcal{N} \leftarrow \mathcal{N}_0$; /* set truncated Taylor series order */
if $(i,j)=(2,1)$ **then**
  $\mathscr{F}_{(i,j)}(t_{\mathrm{f}}) \leftarrow I$; /* set terminal conditions */
else
  $\mathscr{F}_{(i,j)}(t_{\mathrm{f}}) \leftarrow 0$;
end
for $i=0,1,\cdots,\mathcal{N}_0$ **do**
  for $j=0,1,\cdots,i$ **do**
    for $t=t_{\mathrm{f}}$ **to** t_0 **do**
      solve $\dot{\mathscr{F}}_{(i,j)}(t)=-\mathscr{H}_{(i,j)}\left(\mathscr{F}_{(\cdot,\cdot)}(t)\right)$; /* calculate coefficients */
    end
  end
end

Algorithm 3.1: Off-line part, calculate generating function coefficients.

if *there is a computational demand for boundary conditions* $(x_{\text{init}}, x_{\text{term}})$ **then**
  $x(t_0) \leftarrow x_{\text{init}}$, $x(t_{\text{f}}) \leftarrow x_{\text{term}}$; /* set boundary conditions */
  solve $\lambda(t_{\text{f}})$ from $x(t_{\text{f}}) = \left.\frac{\partial F(x,\lambda(t_{\text{f}}),t)}{\partial \lambda(t_{\text{f}})}\right|_{t=t_0}$; /* calculate terminal costate */
  for $t = t_0$ **to** t_{f} **do**
    solve $\dot{x} = Ax - G\frac{\partial F(x,\lambda(t_{\text{f}}),t)}{\partial x}$; /* generate optimal state */
  end
  for $t = t_0$ **to** t_{f} **do**
    $u \leftarrow -R^{-1}B^{\mathsf{T}}\frac{\partial F(x,\lambda(t_{\text{f}}),t)}{\partial x}$; /* generate optimal input */
  end
else
  goto 1; /* on-demand */
end

Algorithm 3.2: On-line part, generate optimal solutions.

In above two algorithms, optimal solutions will be more accurate if we select greater $\mathcal{N}_0$, i.e. expand functions as Taylor series up to higher orders. Since the computation of the coefficients is implemented off-line by Algorithm 3.1, it is free of us to choose any particular orders. From this viewpoint, though the penalized problem is a nonlinear problem, we can still obtain its optimal solutions accurately. On the other side, when we increase $\mathcal{N}_0$, the total number of ordinary differential equations in the off-line part for the coefficients also increases. Therefore, when we select $\mathcal{N}_0$, both the demand of the accuracy and the computational ability of the computer should be taken into account.

3.4 Examples

In this section, we will give two examples. One is to compare results by generating function method with analytic solutions in Section 3.4.1, the other is to illustrate the effectiveness of the generating function method for different boundary conditions by Algorithms 3.1 and 3.2 in Section 3.4.2.

3.4.1 Analytic scalar example

Example 3.1. Consider the minimum energy problem with a second order state variable inequality constraint [10]

$$\min_a \int_0^1 \frac{1}{2}a(t)^2 \mathrm{d}t \tag{3.24}$$

$$\text{s.t. } \dot{v}(t) = a(t), \quad \dot{x}(t) = v(t), \quad t \in [0,1] \tag{3.25}$$

$$v(0) = 1, \quad x(0) = 0, \quad v(1) = -1, \quad x(1) = 0 \tag{3.26}$$

$$x \leqslant 0.1. \tag{3.27}$$

This is a typical path and terminal state constrained problem. For this problem, [10] gives

the exact minimum energy $J^* = 40/9$ and the analytic solutions

$$v(t) = \begin{cases} \left(1 - \dfrac{t}{0.3}\right)^2, & 0 \leqslant t \leqslant 0.3 \\ 0, & 0.3 \leqslant t \leqslant 0.7 \\ -\left(1 - \dfrac{1-t}{0.3}\right)^2, & 0.7 \leqslant t \leqslant 1 \end{cases}, \quad x(t) = \begin{cases} 0.1 - 0.1\left(1 - \dfrac{t}{0.3}\right)^3, & 0 \leqslant t \leqslant 0.3 \\ 0.1, & 0.3 \leqslant t \leqslant 0.7 \\ 0.1 - 0.1\left(1 - \dfrac{1-t}{0.3}\right)^3, & 0.7 \leqslant t \leqslant 1 \end{cases}$$

$$a(t) = \begin{cases} -\dfrac{2}{0.3}\left(1 - \dfrac{t}{0.3}\right), & 0 \leqslant t \leqslant 0.3 \\ 0, & 0.3 \leqslant t \leqslant 0.7 \\ -\dfrac{2}{0.3}\left(1 - \dfrac{1-t}{0.3}\right), & 0.7 \leqslant t \leqslant 1 \end{cases}.$$

To compare with these solutions, we will implement the developed generating function method to solve the problem. First, we design the penalty as

$$P(x) = \frac{1}{0.1 - x} - \frac{x}{0.1^2} \tag{3.28}$$

according to (3.22) based on inverse penalty. Assigning four decreasing values $10^{-1}, 10^{-2}, 10^{-3}$, and 10^{-4} to the factor μ, we show the value of the product $\mu P(x)$ in Figure 3.1. It is easy to find that $\mu P(x)$ with smaller μ comes closer to the boundary of the path constraint 0.1, which should give more accurate solutions than the other greater factors.

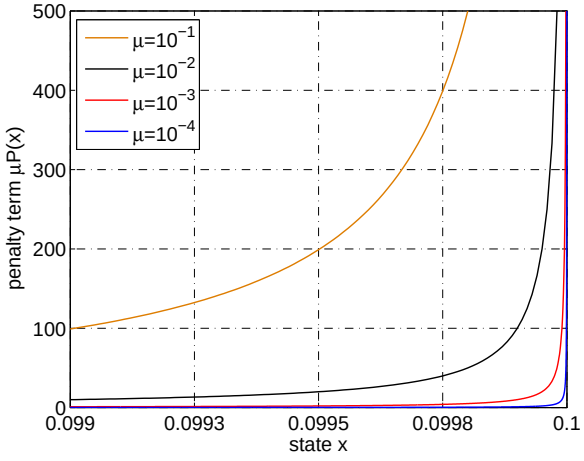
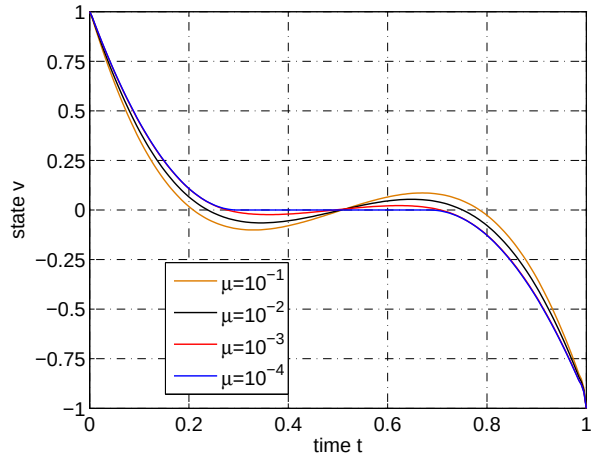


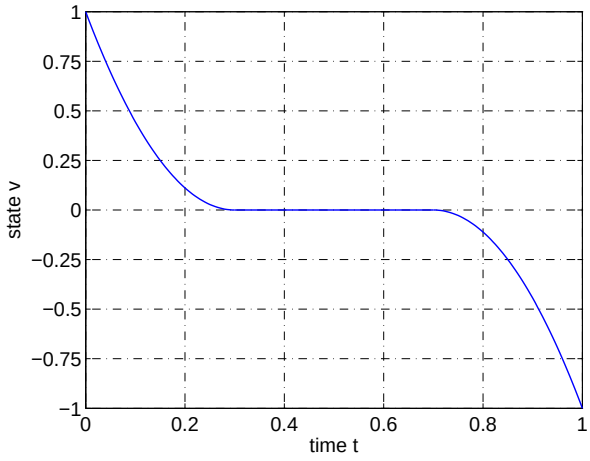
Figure 3.1: Penalty term $\mu P(x)$ with factors $\mu = 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}$.

Table 3.1: Minimum cost function value for $\mu = 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}$, and exact one by comparison.

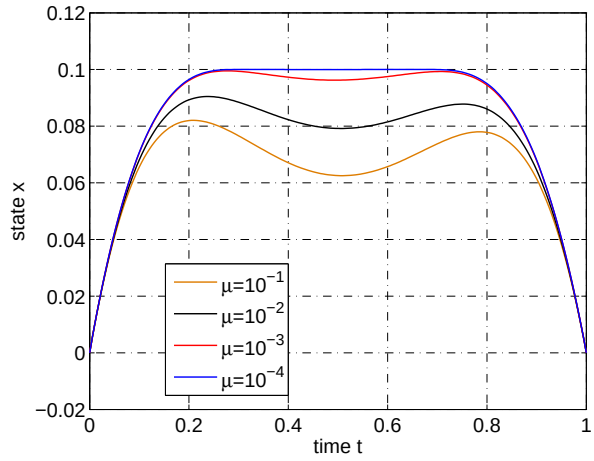
	$\mu = 10^{-1}$	$\mu = 10^{-2}$	$\mu = 10^{-3}$	$\mu = 10^{-4}$	Exact
minimum J	8.0242	5.2206	4.6467	4.6270	4.4444



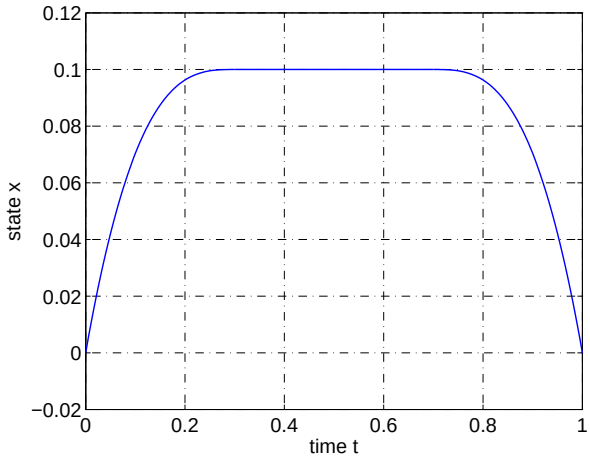
(a) State v generated by generating function method



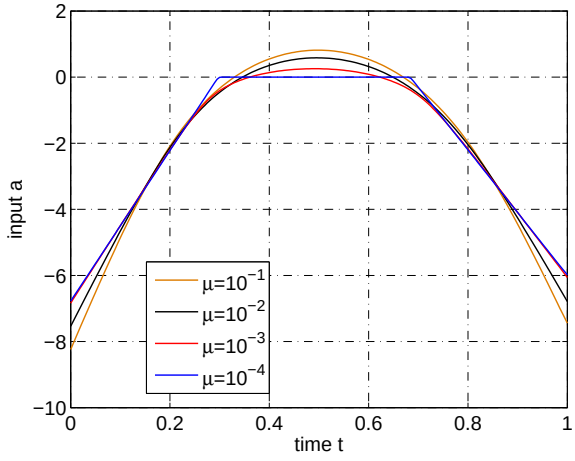
(b) Exact state v



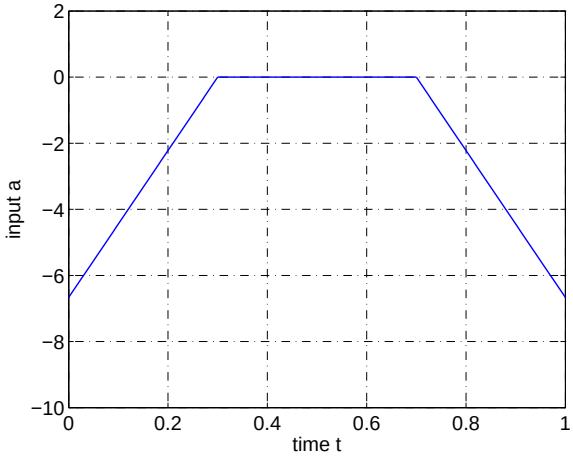
(c) State x generated by generating function method



(d) Exact state x



(e) Input a generated by generating function method



(f) Exact input a

Figure 3.2: Comparison between optimal solutions by generating function method (with $\mu = 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}$) and exact ones.

Second, we write down the Hamilton–Jacobi equation according to (2.33), expand the generating function and the penalty (3.28) in Hamilton–Jacobi equation as Taylor series up to sixth order. Note that the first order Taylor term of the penalty 3.28 is equal to zero, which satisfies the recursive condition and is suitable for the generating function. Due to the Taylor series expansion, we reduce the Hamilton–Jacobi equation (2.33) to ordinary differential equations (3.20). With the boundary conditions (3.21), we can solve the generating function coefficients $\mathscr{F}_{(i,j)}(t)$'s recursively. Finally, by substituting the numerical generating function into (2.42), we obtain the optimal solutions of the penalized problem, which is the approximation of the constrained problem (3.24)–(3.27). We present the results in Figure 3.2 and Table 3.1.

Figure 3.2(a), (c), and (e) are the optimal v, x, and a solved by the generating function method (with factors $\mu = 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}$), respectively. Correspondingly, Figure 3.2(b), (d), and (f) exhibit the analytic solutions. Table 3.1 gives the related minimum cost function values. It can be found from these results that as μ approaches zero, the minimum J and optimal state and input approach the exact ones. This verifies Corollary 3.1 and Theorems 3.2 and 3.3 about the convergence. Moreover, by employing the designed penalty and selecting small factor, the developed generating function method can generate accurate solutions. This demonstrates the effectiveness of the method.

3.4.2 Constrained spacecraft rendezvous

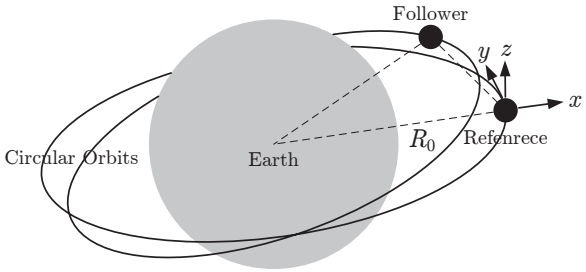


Figure 3.3: Local Vertical Local Horizontal Frame.

The relative orbit between spacecrafts can be described by the Hill–Clohessy–Wiltshire equations [67]. In this model, a so-called reference spacecraft is considered that orbits the Earth in a circular trajectory in Figure 3.3, where $\omega = (\mu_e/R_0^3)^{1/2}$ is the orbit rate, $\mu_e = GM_e$ is the gravitational parameter of the Earth, G is the universal gravitational constant, M_e is the mass of the Earth, and R_0 is the orbital radius of the reference spacecraft (much larger than the relative distance between the spacecrafts). The motion of the follower spacecraft is studied from a reference frame (x, y, z) fixed at center of the reference spacecraft, where x, y, and z are the radial, along-track, and cross-track directions, respectively. This set of coordinate axes is called the Local Vertical Local Horizontal Frame. The relative motion in this frame is given by

$$\begin{aligned}
\ddot{x} =& 2\omega\dot{y} + \omega^2(R_0 + x) - \frac{\mu}{R^3}(R_0 + x) + u_x \\
\ddot{y} =& -2\omega\dot{x} + \omega^2 y - \frac{\mu}{R^3} y + u_y \\
\ddot{z} =& -\frac{\mu}{R^3} z + u_z
\end{aligned}$$

where $R = ((R_0 + x)^2 + y^2 + z^2)^{1/2}$. After nondimensionalization with reference length R_0 and time $1/\omega$, and linearization about $(x, y, z) = (0, 0, 0)$, we have the Hill–Clohessy–Wiltshire equations

$$\begin{aligned}\ddot{x} =& 2\dot{y} + 3x + u_x\\ \ddot{y} =& -2\dot{x} + u_y\\ \ddot{z} =& -z + u_z.\end{aligned}$$

For the sake of simplicity, we only consider the first two in-plane motions (independent of the third out-plane motion)

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\\ \dot{x}_3\\ \dot{x}_4\end{bmatrix} = \begin{bmatrix}0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\\ 3 & 0 & 0 & 2\\ 0 & 0 & -2 & 0\end{bmatrix}\begin{bmatrix}x_1\\ x_2\\ x_3\\ x_4\end{bmatrix} + \begin{bmatrix}0 & 0\\ 0 & 0\\ 1 & 0\\ 0 & 1\end{bmatrix}\begin{bmatrix}u_1\\ u_2\end{bmatrix} \tag{3.29}$$

where $[x_1, x_2, x_3, x_4]^\mathsf{T} = [x, y, \dot{x}, \dot{y}]^\mathsf{T} = [x, y, v_x, v_y]^\mathsf{T}$ and $[u_1, u_2]^\mathsf{T} = [u_x, u_y]^\mathsf{T}$.

Consider the follower spacecraft satisfies the dynamics (3.29) with the specified initial state boundary conditions, and transits to the origin (reference) in fixed amount of time $[t_0, t_\mathrm{f}]$. Our objective is to find optimal input to minimize the energy considered cost function

$$J = \frac{1}{2}\int_{t_0}^{t_\mathrm{f}} u^\mathsf{T} u \mathrm{d}t.$$

This is the optimal rendezvous problem. More generally, we should consider the obstacle avoidance problem for the spacecraft, and also the velocity limit during transitions. All these cases can be treated as the state constraints.

We set the example from [52]: the follower spacecraft starts from the initial positions locating along the radius 0.15 and velocities identically zero (specifically $[0.15\cos\theta, 0.15\sin\theta, 0, 0]^\mathsf{T}$ with θ varying from 0 to 2π by the step $\pi/8$), transits to the origin $[0, 0, 0, 0]^\mathsf{T}$ in one unit time. Additionally, we also consider the velocity constraints[†]

$$-0.2 \leqslant v_x \leqslant 0.2, \quad -0.2 \leqslant v_y \leqslant 0.2. \tag{3.30}$$

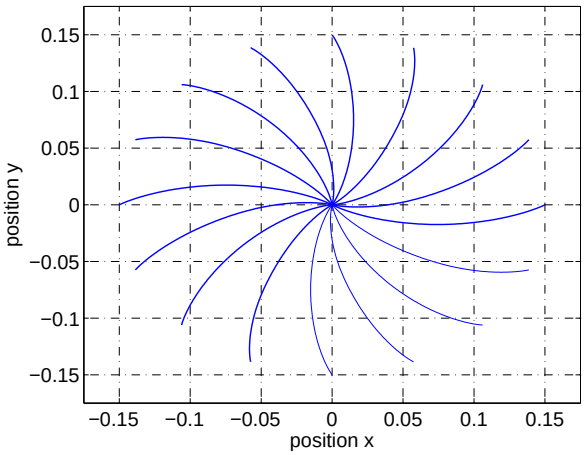
We apply Algorithms 3.1 and 3.2 to this velocity constrained rendezvous problem. For the constraints (3.30), we design the penalty

$$\frac{1}{v_x + 0.2} + \frac{1}{0.2 - v_x} + \frac{1}{v_y + 0.2} + \frac{1}{0.2 - v_y}$$

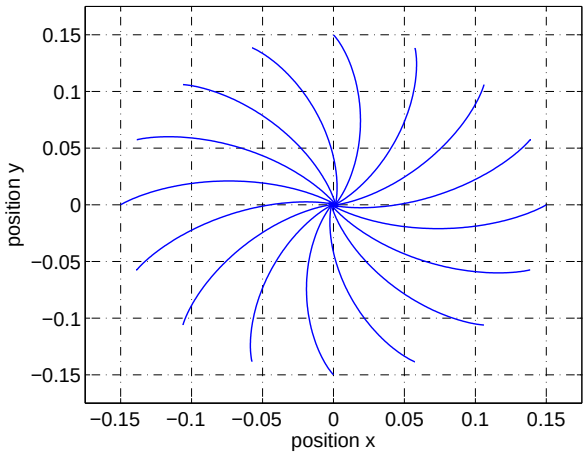
and select the penalty factor 10^{-6}. In the off-line part, we expand the functions as Taylor series up to sixth orders, calculate and save the generating function coefficients in advance. During on-line computations, we read these coefficients to efficiently generate trajectories for different specified boundary conditions.

[†]Due to the formulation of Problem 3.1, various of constraints, e.g. velocity limits or position obstacles or these two mixed constraints and so on, can be well tackled by the developed method for the spacecraft, here we only set the velocity limits example.

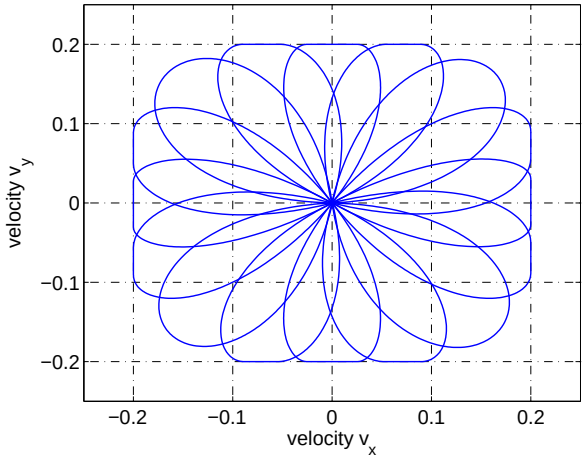
Results are presented in Figure 3.4, where figures in the left column are the position and velocity trajectories for the above constrained problem, while figures right column are the results in [52] for the problem without velocity limits (3.30). The off-line and on-line computational time is 0.0054 [s] and 0.0037 [s] according to Algorithms 3.1 and 3.2, respectively. These results well demonstrate the computational efficiency of the generating function method for different boundary conditions, especially the number of the boundary condition is large. Note that this advantage is not only limited to different boundary conditions, but also different time intervals. Furthermore, by the comparison, we successfully extend the generating function method to state constrained problems and well solve this constrained rendezvous application problem.



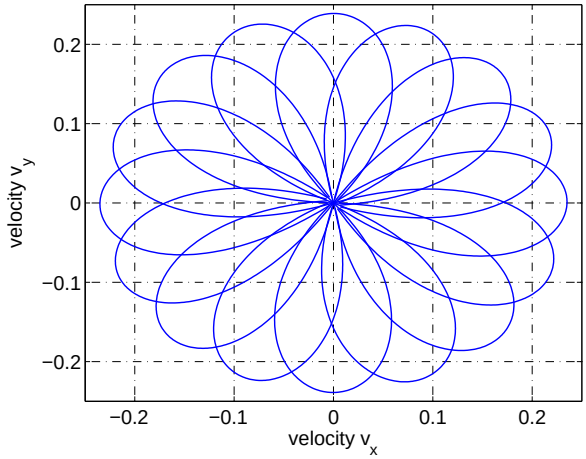
(a) Constrained position trajectories



(b) Unconstrained position trajectories



(c) Constrained velocity trajectories



(d) Unconstrained velocity trajectories

Figure 3.4: Comparison between optimal constrained trajectories (by the developed method) and unconstrained trajectories (from [52]).

3.5 Summary

This chapter extends the generating function method to the LQ optimal control problems with path and terminal state constraints by employing penalties. The penalized problem with a general penalty is introduced to approximate the original constrained problem. We show that both of them are convex problems and optimal solutions of penalized problem will converge to the ones of original constrained problem when the penalty factor approaches zero. Moreover, a recursive condition is presented to eliminate the coupling relation between the generating function coefficients with lower and higher indices in the ordinary differential equations so that they can be solved recursively. This finally enables us to solve the penalized problem by generating function method. Based on this, we summarize how to design penalties that is suitable for the generating function method, and give an algorithm presents how to generate optimal solutions repetitively for different boundary conditions. This framework is able to give accurate solutions, and also possesses the significance in online repetitive computation for different boundary conditions. Examples illustrate the effectiveness of the developed method.

Chapter 4

Discrete-time LQ optimal control problem

For the case of continuous-time problems, [36, 37] use only one generating function[†] to generate optimal solutions. This method gives the optimal input as state feedback control, and enables us to calculate generating function coefficients off-line in advance, and to generate optimal solutions by integrating the system equation on-line. From this viewpoint, it is useful for on-line repetitive computation of optimal solutions satisfying different boundary conditions. In order to further reduce the on-line computational effort, the double generating functions method is proposed [39]. Compared with single method, the double generating functions method gives the optimal solutions as algebraic expressions in terms of pre-computed coefficients and boundary conditions based on a pair of different generating functions. Hence in on-line computation, we only need to read saved coefficients and each set of boundary conditions to generate optimal solutions by algebraic manipulations without integration of the system equation. This method doubles the off-line work, but is more efficient in on-line computation. For the case of discrete-time problems, there is only one paper [50] that develops the single generating function method for the LQ optimal control problem.

Further, by careful investigating the generators developed by single/double generating function(s) method, we will find there exist inverse terms. If the singularity would occur at some time steps or periods, it will cause the serious numerical instabilities. Therefore, the invertibility analysis of the inverse terms should be preformed to help us select the numerical stable generators. So far, only [39] has given some preliminary analysis on this issue to show that the developed generators for optimal solutions constructed by double generating functions with the same time directions will cause instabilities when the time interval increases.

This chapter develops the discrete analogue of double generating functions method. To clearly present the fundamental feature of this method and to make it convenient for the further extension to nonlinear problems, this chapter investigates the classical discrete-time LQ optimal control problem. First in Section 4.1, we derive the left discrete Hamiltonian, Hamilton's equa-

[†]The reference chapters [36, 37] use the generating function $F_{1\mathrm{f}}$ to generate optimal solutions. Before the solution generation, since $F_{1\mathrm{f}}$ is not well-defined at initial time, [36, 37] employ the Legendre transformation to obtain $F_{1\mathrm{f}}$ from $F_{2\mathrm{f}}$ at first. Though two generating functions are used in this framework in fact, it is still called as single generating function method.

tions, and the Hamilton–Jacobi equation for the LQ optimal control problem which is a counterpart to the right ones in [50] according to discrete mechanics [48]. Second in Section 4.2, we choose appropriate Hamilton–Jacobi equation, left or right, to solve for the forward type II, III, and backward type III generating functions. Then by selecting any two different generating functions from the four single ones, we have six pairs of generating functions which give six generators for optimal solutions, respectively. These discrete generators maintain the advantage of on-line efficient computation for different boundary conditions, which is presented by a followed algorithm. Besides, since each generator contains inverse terms, we deeply perform the invertibility analysis in Section 4.3 to conclude that the terms in the generators constructed by double generating functions with opposite time directions are invertible under some mild conditions, while the terms with the same time directions will become singular when the time goes infinity which may cause instabilities in numerical computations. Examples in Section 4.4 illustrate the effectiveness of the developed method. Section 4.5 summarizes this chapter.

4.1 Problem setting and necessary conditions for optimality

This section presents the problem setting in Section 4.1.1, and the necessary conditions for optimality in terms of right and left discrete Hamiltonians in Section 4.1.2.

4.1.1 Problem setting

In this chapter, we study the discrete-time LQ optimal control problem, i.e. Problem 2.4 formulated in Section 2.2.4. We here present it again in the following to make this chapter self-contained and convenient for the reading.

Problem 4.1.

$$\min_{u} \sum_{k=0}^{N-1} \frac{1}{2} \left(x_k^{\mathsf{T}} Q x_k + u_k^{\mathsf{T}} R u_k \right) \tag{4.1}$$

$$\text{s.t. } x_{k+1} = A x_k + B u_k, \quad k = 0, 1, \cdots, N-1 \tag{4.2}$$

$$x_0 = x_{\text{init}}, \quad x_N = x_{\text{term}} \tag{4.3}$$

where the constant matrices $Q \in \mathbb{R}^{n \times n}$, $R \in \mathbb{R}^{m \times m}$, $A \in \mathbb{R}^{n \times n}$, and $B \in \mathbb{R}^{n \times m}$. Moreover, the matrices $Q \succcurlyeq 0$, $R \succ 0$, and A is invertible.

4.1.2 Necessary conditions for optimality

According to Theorem 2.6, the necessary conditions for optimizing Problem 4.1 represented by the right discrete Hamiltonian is

$$x_{k+1} = \frac{\partial H^+(x_k, \lambda_{k+1})}{\partial \lambda_{k+1}} \tag{4.4}$$

$$\lambda_k = \frac{\partial H^+(x_k, \lambda_{k+1})}{\partial x_k} \tag{4.5}$$

$$u_k = -M\lambda_{k+1} \tag{4.6}$$

where the right discrete Hamiltonian

$$H^+(x_k, \lambda_{k+1}) = \frac{1}{2}x_k^\mathsf{T} Q x_k + \lambda_{k+1}^\mathsf{T} A x_k - \frac{1}{2}\lambda_{k+1}^\mathsf{T} G \lambda_{k+1}$$

the matrices $M := R^{-1}B^\mathsf{T}$, and $G := BR^{-1}B^\mathsf{T} \succcurlyeq 0$.

On the other hand, the necessary conditions for optimality can also be represented by the left discrete Hamiltonian

$$x_k = -\frac{\partial H^-(\lambda_k, x_{k+1})}{\partial \lambda_k} \tag{4.7}$$

$$\lambda_{k+1} = -\frac{\partial H^-(\lambda_k, x_{k+1})}{\partial x_{k+1}} \tag{4.8}$$

$$u_k = -M\lambda_{k+1}$$

where the left discrete Hamiltonian

$$H^-(\lambda_k, x_{k+1}) := -\frac{1}{2}x_{k+1}^\mathsf{T}\mathcal{Q}x_{k+1} - \lambda_k^\mathsf{T}\mathcal{A}x_{k+1} + \frac{1}{2}\lambda_k^\mathsf{T}\mathcal{G}\lambda_k.$$

Here, we will derive the expression of H^-, i.e. expressions for the unknown matrices $\mathcal{A}$, $\mathcal{Q}$, and $\mathcal{G} \in \mathbb{R}^{n\times n}$, through Legendre transformation (2.119) from H^+

$$H^-(\lambda_k, x_{k+1}) = H^+(x_k, \lambda_{k+1}) - \lambda_k^\mathsf{T} x_k - \lambda_{k+1}^\mathsf{T} x_{k+1}.$$

After substitution of (4.7) into the above Legendre transformation, we have

$$\begin{aligned}
\mathcal{A} =& (A + GA^{-\mathsf{T}}Q)^{-1} \\
\mathcal{Q} =& -(A^\mathsf{T} + QA^{-1}G)^{-1}(QA^{-1}GA^{-\mathsf{T}}Q + Q)(A + GA^{-\mathsf{T}}Q)^{-1} \\
\mathcal{G} =& -(A + GA^{-\mathsf{T}}Q)^{-1}(GA^{-\mathsf{T}}QA^{-1}G + G)(A^\mathsf{T} + QA^{-1}G)^{-1}.
\end{aligned}$$

Remark 4.1. The above three expressions need the formula $A + GA^{-\mathsf{T}}Q$ (and its transpose) to be invertible. This can be proven by the following: $\exists y \in \mathbb{R}^n$ such that

$$\begin{aligned}
& (A + GA^{-\mathsf{T}}Q)y = 0 \\
\Rightarrow\ & y + A^{-1}GA^{-\mathsf{T}}Qy = 0 \qquad (4.9) \\
\Rightarrow\ & Qy + QA^{-1}GA^{-\mathsf{T}}Qy = 0 \\
\Rightarrow\ & (I + QA^{-1}GA^{-\mathsf{T}}Q)Qy = 0 \\
\Rightarrow\ & Qy = 0. \qquad (4.10)
\end{aligned}$$

Combining (4.9) and (4.10), we get $y = 0$ which implies that $A + GA^{-\mathsf{T}}Q$ is nonsingular. Moreover, the fact $\mathcal{Q} \preccurlyeq 0$ and $\mathcal{G} \preccurlyeq 0$ can also be verified. First, we know that $QA^{-1}GA^{-\mathsf{T}}Q \succcurlyeq 0$ by the definition of positive semi-definite. Further, it is known that the sum of two positive semi-definite matrices $QA^{-1}GA^{-\mathsf{T}}Q + Q \succcurlyeq 0$. Based on these, we have $\mathcal{Q} = -(A^\mathsf{T} + QA^{-1}G)^{-1}(QA^{-1}GA^{-\mathsf{T}}Q + Q)(A + GA^{-\mathsf{T}}Q)^{-1} \preccurlyeq 0$ also by the definition. The result $\mathcal{G} = -(A + GA^{-\mathsf{T}}Q)^{-1}(GA^{-\mathsf{T}}QA^{-1}G + G)(A^\mathsf{T} + QA^{-1}G)^{-1} \preccurlyeq 0$ can also be obtained by the similar way.

In this chapter, we make a standard assumption that (A, G) and $(\mathcal{A}, \mathcal{G})$ are controllable, (Q, A) and $(\mathcal{Q}, \mathcal{A})$ are observable. Note that (A, G) controllable and $(\mathcal{Q}, \mathcal{A})$ observable are equivalent, while $(\mathcal{A}, \mathcal{G})$ controllable and (Q, A) observable are equivalent.

The two sets of necessary conditions in this subsection are equivalent, and are the bases for the next section. The right Hamilton's equations (4.4) with boundary conditions (4.3), or the left Hamilton's equations (4.7) with boundary conditions (4.3) compose the two point boundary value problem. Evaluating the optimal trajectory of Problem 4.1 corresponds to solving the two point boundary value problem. The double generating functions method will be developed in the remainder to solve this problem.

4.2 Double generating functions method

In this section, We give exact expressions of the generating functions $F_{2\mathrm{f}}$, $F_{3\mathrm{f}}$, $F_{2\mathrm{b}}$, and $F_{3\mathrm{b}}$ in Section 4.2.1. Based on this, we finally give six generators for optimal solutions only in terms of pre-computed coefficients and boundary conditions by six different pairs of generating functions, respectively in Section 4.2.2.

4.2.1 Generating functions

Since T. Lee [50] only gives the exact expression of $F_{2\mathrm{b}}$ in Proposition 2.4, we here derive the other three generating functions $F_{2\mathrm{f}}$, $F_{3\mathrm{f}}$, and $F_{3\mathrm{b}}$ in the following proposition (also present $F_{2\mathrm{b}}$).

Proposition 4.1. *For Problem 4.1*

(i) *The generating function $F_{2\mathrm{f}}(x_k, \lambda_0, k)$ has the expression of*

$$F_{2\mathrm{f}}(x_k, \lambda_0, k) = \frac{1}{2} x_k^{\mathsf{T}} \mathscr{U}_{2\mathrm{f},k} x_k + \lambda_0^{\mathsf{T}} \mathscr{V}_{2\mathrm{f},k} x_k + \frac{1}{2} \lambda_0^{\mathsf{T}} \mathscr{W}_{2\mathrm{f},k} \lambda_0 \tag{4.11}$$

where the coefficients $\mathscr{U}_{2\mathrm{f},k} = \mathscr{U}_{2\mathrm{f},k}^{\mathsf{T}} \in \mathbb{R}^{n\times n}$, $\mathscr{V}_{2\mathrm{f},k} \in \mathbb{R}^{n\times n}$, and $\mathscr{W}_{2\mathrm{f},k} = \mathscr{U}_{2\mathrm{f},k}^{\mathsf{T}} \in \mathbb{R}^{n\times n}$ are the solutions of the difference equations ($k = 0, 1, \cdots, N-1$)

$$\mathscr{U}_{2\mathrm{f},k+1} = \mathcal{A}^{\mathsf{T}} (I + \mathscr{U}_{2\mathrm{f},k} \mathcal{G})^{-1} \mathscr{U}_{2\mathrm{f},k} \mathcal{A} + \mathcal{Q} \tag{4.12}$$

$$\mathscr{V}_{2\mathrm{f},k+1} = \mathscr{V}_{2\mathrm{f},k} (I + \mathcal{G} \mathscr{U}_{2\mathrm{f},k})^{-1} \mathcal{A} \tag{4.13}$$

$$\mathscr{W}_{2\mathrm{f},k+1} = \mathscr{W}_{2\mathrm{f},k} - \mathscr{V}_{2\mathrm{f},k} (I + \mathcal{G} \mathscr{U}_{2\mathrm{f},k})^{-1} \mathcal{G} \mathscr{V}_{2\mathrm{f},k}^{\mathsf{T}} \tag{4.14}$$

with the boundary conditions $\mathscr{U}_{2\mathrm{f},0} = 0$, $\mathscr{V}_{2\mathrm{f},0} = I$, and $\mathscr{W}_{2\mathrm{f},0} = 0$.

(ii) *The generating function $F_{3\mathrm{f}}(\lambda_k, x_0, k)$ has the expression of*

$$F_{3\mathrm{f}}(\lambda_k, x_0, k) = \frac{1}{2} \lambda_k^{\mathsf{T}} \mathscr{U}_{3\mathrm{f},k} \lambda_k + x_0^{\mathsf{T}} \mathscr{V}_{3\mathrm{f},k} \lambda_k + \frac{1}{2} x_0^{\mathsf{T}} \mathscr{W}_{3\mathrm{f},k} x_0 \tag{4.15}$$

where the coefficients $\mathscr{U}_{3\mathrm{f},k} = \mathscr{U}_{3\mathrm{f},k}^{\mathsf{T}} \in \mathbb{R}^{n\times n}$, $\mathscr{V}_{3\mathrm{f},k} \in \mathbb{R}^{n\times n}$, and $\mathscr{W}_{3\mathrm{f},k} = \mathscr{U}_{3\mathrm{f},k}^{\mathsf{T}} \in \mathbb{R}^{n\times n}$ are the solutions of the difference equations ($k = 0, 1, \cdots, N-1$)

$$\mathscr{U}_{3\mathrm{f},k+1} = A (I + \mathscr{U}_{3\mathrm{f},k} Q)^{-1} \mathscr{U}_{3\mathrm{f},k} A^{\mathsf{T}} + G \tag{4.16}$$

$$\mathscr{V}_{3\mathrm{f},k+1} = \mathscr{V}_{3\mathrm{f},k}(I + Q\mathscr{U}_{3\mathrm{f},k})^{-1}A^{\mathsf{T}} \tag{4.17}$$

$$\mathscr{W}_{3\mathrm{f},k+1} = \mathscr{W}_{3\mathrm{f},k} - \mathscr{V}_{3\mathrm{f},k}(I + Q\mathscr{U}_{3\mathrm{f},k})^{-1}Q\mathscr{V}_{3\mathrm{f},k}^{\mathsf{T}} \tag{4.18}$$

with the boundary conditions $\mathscr{U}_{3\mathrm{f},0} = 0$, $\mathscr{V}_{3\mathrm{f},0} = -I$, *and* $\mathscr{W}_{3\mathrm{f},0} = 0$.

(iii) *The generating function* $F_{2\mathrm{b}}(x_k, \lambda_N, k)$ *has the expression of*

$$F_{2\mathrm{b}}(x_k, \lambda_N, k) = \frac{1}{2}x_k^{\mathsf{T}}\mathscr{U}_{2\mathrm{b},\kappa}x_k + \lambda_N^{\mathsf{T}}\mathscr{V}_{2\mathrm{b},\kappa}x_k + \frac{1}{2}\lambda_N^{\mathsf{T}}\mathscr{W}_{2\mathrm{b},\kappa}\lambda_N \tag{4.19}$$

where $\kappa = k - N$[†], *and the coefficients* $\mathscr{U}_{2\mathrm{b},\kappa} = \mathscr{U}_{2\mathrm{b},\kappa}^{\mathsf{T}} \in \mathbb{R}^{n\times n}$, $\mathscr{V}_{2\mathrm{b},\kappa} \in \mathbb{R}^{n\times n}$, *and* $\mathscr{W}_{2\mathrm{b},\kappa} = \mathscr{U}_{2\mathrm{b},\kappa}^{\mathsf{T}} \in \mathbb{R}^{n\times n}$ *are the solutions of the difference equations* ($\kappa = 0, -1, \cdots, -N+1$)

$$\mathscr{U}_{2\mathrm{b},\kappa-1} = A^{\mathsf{T}}(I + \mathscr{U}_{2\mathrm{b},\kappa}G)^{-1}\mathscr{U}_{2\mathrm{b},\kappa}A + Q \tag{4.20}$$

$$\mathscr{V}_{2\mathrm{b},\kappa-1} = \mathscr{V}_{2\mathrm{b},\kappa}(I + G\mathscr{U}_{2\mathrm{b},\kappa})^{-1}A \tag{4.21}$$

$$\mathscr{W}_{2\mathrm{b},\kappa-1} = \mathscr{W}_{2\mathrm{b},\kappa} - \mathscr{V}_{2\mathrm{b},\kappa}(I + G\mathscr{U}_{2\mathrm{b},\kappa})^{-1}G\mathscr{V}_{2\mathrm{b},\kappa}^{\mathsf{T}} \tag{4.22}$$

with boundary conditions $\mathscr{U}_{2\mathrm{b},0} = 0$, $\mathscr{V}_{2\mathrm{b},0} = I$, *and* $\mathscr{W}_{2\mathrm{b},0} = 0$.

(iv) *The generating function* $F_{3\mathrm{b}}(\lambda_k, x_N, k)$ *has the expression of*

$$F_{3\mathrm{b}}(\lambda_k, x_N, k) = \frac{1}{2}\lambda_k^{\mathsf{T}}\mathscr{U}_{3\mathrm{b},\kappa}\lambda_k + x_N^{\mathsf{T}}\mathscr{V}_{3\mathrm{b},\kappa}\lambda_k + \frac{1}{2}x_N^{\mathsf{T}}\mathscr{W}_{3\mathrm{b},\kappa}x_N \tag{4.23}$$

where the coefficients $\mathscr{U}_{3\mathrm{b},\kappa} = \mathscr{U}_{3\mathrm{b},\kappa}^{\mathsf{T}} \in \mathbb{R}^{n\times n}$, $\mathscr{V}_{3\mathrm{b},\kappa} \in \mathbb{R}^{n\times n}$, *and* $\mathscr{W}_{3\mathrm{b},\kappa} = \mathscr{U}_{3\mathrm{b},\kappa}^{\mathsf{T}} \in \mathbb{R}^{n\times n}$ *are the solutions of the difference equations* ($\kappa = 0, -1, \cdots, -N+1$)

$$\mathscr{U}_{3\mathrm{b},\kappa-1} = \mathcal{A}(I + \mathscr{U}_{3\mathrm{b},\kappa}\mathcal{Q})^{-1}\mathscr{U}_{3\mathrm{b},\kappa}\mathcal{A}^{\mathsf{T}} + \mathcal{G} \tag{4.24}$$

$$\mathscr{V}_{3\mathrm{b},\kappa-1} = \mathscr{V}_{3\mathrm{b},\kappa}(I + \mathcal{Q}\mathscr{U}_{3\mathrm{b},\kappa})^{-1}\mathcal{A}^{\mathsf{T}} \tag{4.25}$$

$$\mathscr{W}_{3\mathrm{b},\kappa-1} = \mathscr{W}_{3\mathrm{b},\kappa} - \mathscr{V}_{3\mathrm{b},\kappa}(I + \mathcal{Q}\mathscr{U}_{3\mathrm{b},\kappa})^{-1}\mathcal{Q}\mathscr{V}_{3\mathrm{b},\kappa}^{\mathsf{T}} \tag{4.26}$$

with boundary conditions $\mathscr{U}_{3\mathrm{b},0} = 0$, $\mathscr{V}_{3\mathrm{b},0} = -I$, *and* $\mathscr{W}_{3\mathrm{b},0} = 0$.

Proof. (i) First, it is known that $F_{2\mathrm{f}}$ is in quadratic form as (4.11) [35]. Then from Remark 2.7, we have $F_{2\mathrm{f}}(x_k, \lambda_0, k)|_{k=0} = \lambda_0^{\mathsf{T}}x_0$ that gives the boundary conditions $\mathscr{U}_{2\mathrm{f},0} = 0$, $\mathscr{V}_{2\mathrm{f},0} = I$, and $\mathscr{W}_{2\mathrm{f},0} = 0$. We solve the Hamilton–Jacobi equation (2.121) to obtain the explicit expression of $F_{2\mathrm{f}}(x_k, \lambda_0, k)$, i.e. difference equations for its coefficient matrices $\mathscr{U}_{2\mathrm{f},k}$, $\mathscr{V}_{2\mathrm{f},k}$, and $\mathscr{W}_{2\mathrm{f},k}$. We first rewrite (2.121) to be an equation only in terms of x_{k+1} and λ_0 by the help of (4.7) and (2.99). Then, since this equation should be satisfied for any x_{k+1} and λ_0, their coefficients can only be zero which leads to the difference equations (4.12)–(4.14). Further, it is known that both $\mathscr{U}_{2\mathrm{f},k}$ and $\mathscr{W}_{2\mathrm{f},k}$ are symmetric due to (4.12) and (4.14) with zero initial conditions $\mathscr{U}_{2\mathrm{f},0} = 0$ and $\mathscr{W}_{2\mathrm{f},0} = 0$.

(ii)–(iv) The exact expressions of $F_{3\mathrm{f}}$ and $F_{3\mathrm{b}}$ can be obtained by the similar way as for $F_{2\mathrm{f}}$ above by solving the Hamilton–Jacobi equations (2.101) and (2.126), respectively. The exact expression of $F_{2\mathrm{b}}$ refers to [50]. □

[†]To make it convenient for the invertibility analysis in the next section, we here transform the time steps of backward generating function coefficients from $N, N-1, \cdots, 0$ to $0, -1, \cdots, -N$ by defining $\kappa = k - N$.

4.2.2 Optimal solutions via Double Generating Functions

The boundary conditions of the state, x_0 and x_N, are pre-given. Dually, the boundary conditions of the costate λ_0 and λ_N, which are required in the next theorem, can be derived by letting $k = 0$ in (2.114) and $k = N$ in (2.102), respectively as

$$\lambda_0 = -\mathscr{U}_{3\mathrm{b},-N}^{-1}(x_0 + \mathscr{V}_{3\mathrm{b},-N}^{\mathsf{T}} x_N) \tag{4.27}$$

$$\lambda_N = -\mathscr{U}_{3\mathrm{f},N}^{-1}(x_N + \mathscr{V}_{3\mathrm{f},N}^{\mathsf{T}} x_0) \tag{4.28}$$

or by letting $k = N$ in (2.100) and $k = 0$ in (2.112), respectively as

$$\lambda_0 = \mathscr{W}_{2\mathrm{f},N}^{-1}(x_0 - \mathscr{V}_{2\mathrm{f},N} x_N) \tag{4.29}$$

$$\lambda_N = \mathscr{W}_{2\mathrm{b},-N}^{-1}(x_N - \mathscr{V}_{2\mathrm{b},-N} x_0). \tag{4.30}$$

Six kinds of double generating functions can be constructed by selecting any two different single generating functions among $F_{2\mathrm{f}}$, $F_{3\mathrm{f}}$, $F_{2\mathrm{b}}$, and $F_{3\mathrm{b}}$. Based on this, we give six generators correspondingly for optimal solutions only in terms of pre-computed coefficients and boundary conditions by the following theorem.

Theorem 4.1. *The optimal state x_k^* and input u_k^* of Problem 4.1 are given as*

$$\begin{bmatrix} x_k^* \\ u_k^* \end{bmatrix} = \begin{bmatrix} \mathscr{U}_{3\mathrm{b},\kappa}(\mathscr{U}_{3\mathrm{f},k} - \mathscr{U}_{3\mathrm{b},\kappa})^{-1}\mathscr{V}_{3\mathrm{f},k}^{\mathsf{T}} & -\mathscr{U}_{3\mathrm{f},k}(\mathscr{U}_{3\mathrm{f},k} - \mathscr{U}_{3\mathrm{b},\kappa})^{-1}\mathscr{V}_{3\mathrm{b},\kappa}^{\mathsf{T}} \\ M(\mathscr{U}_{3\mathrm{f},k+1} - \mathscr{U}_{3\mathrm{b},\kappa+1})^{-1}\mathscr{V}_{3\mathrm{f},k+1}^{\mathsf{T}} & -M(\mathscr{U}_{3\mathrm{f},k+1} - \mathscr{U}_{3\mathrm{b},\kappa+1})^{-1}\mathscr{V}_{3\mathrm{b},\kappa+1}^{\mathsf{T}} \end{bmatrix} \begin{bmatrix} x_0 \\ x_N \end{bmatrix} \tag{4.31}$$

or

$$\begin{bmatrix} x_k^* \\ u_k^* \end{bmatrix} = \begin{bmatrix} -(I + \mathscr{U}_{3\mathrm{f},k}\mathscr{U}_{2\mathrm{b},\kappa})^{-1}\mathscr{V}_{3\mathrm{f},k}^{\mathsf{T}} & -(I + \mathscr{U}_{3\mathrm{f},k}\mathscr{U}_{2\mathrm{b},\kappa})^{-1}\mathscr{U}_{3\mathrm{f},k}\mathscr{V}_{2\mathrm{b},\kappa}^{\mathsf{T}} \\ M(I + \mathscr{U}_{2\mathrm{b},\kappa+1}\mathscr{U}_{3\mathrm{f},k+1})^{-1}\mathscr{U}_{2\mathrm{b},\kappa+1}\mathscr{V}_{3\mathrm{f},k+1}^{\mathsf{T}} & -M(I + \mathscr{U}_{2\mathrm{b},\kappa+1}\mathscr{U}_{3\mathrm{f},k+1})^{-1}\mathscr{V}_{2\mathrm{b},\kappa+1}^{\mathsf{T}} \end{bmatrix} \begin{bmatrix} x_0 \\ \lambda_N \end{bmatrix} \tag{4.32}$$

where λ_N by (4.28), or

$$\begin{bmatrix} x_k^* \\ u_k^* \end{bmatrix} = \begin{bmatrix} -(I + \mathscr{U}_{3\mathrm{b},\kappa}\mathscr{U}_{2\mathrm{f},k})^{-1}\mathscr{U}_{3\mathrm{b},\kappa}\mathscr{V}_{2\mathrm{f},k}^{\mathsf{T}} & -(I + \mathscr{U}_{3\mathrm{b},\kappa}\mathscr{U}_{2\mathrm{f},k})^{-1}\mathscr{V}_{3\mathrm{b},\kappa}^{\mathsf{T}} \\ -M(I + \mathscr{U}_{2\mathrm{f},k+1}\mathscr{U}_{3\mathrm{b},\kappa+1})^{-1}\mathscr{V}_{2\mathrm{f},k+1}^{\mathsf{T}} & M(I + \mathscr{U}_{2\mathrm{f},k+1}\mathscr{U}_{3\mathrm{b},\kappa+1})^{-1}\mathscr{U}_{2\mathrm{f},k+1}\mathscr{V}_{3\mathrm{b},\kappa+1}^{\mathsf{T}} \end{bmatrix} \begin{bmatrix} \lambda_0 \\ x_N \end{bmatrix} \tag{4.33}$$

where λ_0 by (4.27), or

$$\begin{bmatrix} x_k^* \\ u_k^* \end{bmatrix} = \begin{bmatrix} (\mathscr{U}_{2\mathrm{b},\kappa} - \mathscr{U}_{2\mathrm{f},k})^{-1}\mathscr{V}_{2\mathrm{f},k}^{\mathsf{T}} & -(\mathscr{U}_{2\mathrm{b},\kappa} - \mathscr{U}_{2\mathrm{f},k})^{-1}\mathscr{V}_{2\mathrm{b},\kappa}^{\mathsf{T}} \\ -M\mathscr{U}_{2\mathrm{b},\kappa+1}(\mathscr{U}_{2\mathrm{b},\kappa+1} - \mathscr{U}_{2\mathrm{f},k+1})^{-1}\mathscr{V}_{2\mathrm{f},k+1}^{\mathsf{T}} & M\mathscr{U}_{2\mathrm{f},k+1}(\mathscr{U}_{2\mathrm{b},\kappa+1} - \mathscr{U}_{2\mathrm{f},k+1})^{-1}\mathscr{V}_{2\mathrm{b},\kappa+1}^{\mathsf{T}} \end{bmatrix} \begin{bmatrix} \lambda_0 \\ \lambda_N \end{bmatrix} \tag{4.34}$$

where λ_0 by (4.29) and λ_N by (4.30), or

$$\begin{bmatrix} x_k^* \\ u_k^* \end{bmatrix} = \begin{bmatrix} -(I + \mathscr{U}_{3\mathrm{f},k}\mathscr{U}_{2\mathrm{f},k})^{-1}\mathscr{V}_{3\mathrm{f},k}^{\mathsf{T}} & -(I + \mathscr{U}_{3\mathrm{f},k}\mathscr{U}_{2\mathrm{f},k})^{-1}\mathscr{U}_{3\mathrm{f},k}\mathscr{V}_{2\mathrm{f},k}^{\mathsf{T}} \\ M(I + \mathscr{U}_{2\mathrm{f},k+1}\mathscr{U}_{3\mathrm{f},k+1})^{-1}\mathscr{U}_{2\mathrm{f},k+1}\mathscr{V}_{3\mathrm{f},k+1}^{\mathsf{T}} & -M(I + \mathscr{U}_{2\mathrm{f},k+1}\mathscr{U}_{3\mathrm{f},k+1})^{-1}\mathscr{V}_{2\mathrm{f},k+1}^{\mathsf{T}} \end{bmatrix} \begin{bmatrix} x_0 \\ \lambda_0 \end{bmatrix} \tag{4.35}$$

where λ_0 by (4.29), or

$$\begin{bmatrix} x_k^* \\ u_k^* \end{bmatrix} = \begin{bmatrix} -(I + \mathscr{U}_{3\mathrm{b},\kappa}\mathscr{U}_{2\mathrm{b},\kappa})^{-1}\mathscr{V}_{3\mathrm{b},\kappa}^{\mathsf{T}} & -(I + \mathscr{U}_{3\mathrm{b},\kappa}\mathscr{U}_{2\mathrm{b},\kappa})^{-1}\mathscr{U}_{3\mathrm{b},\kappa}\mathscr{V}_{2\mathrm{b},\kappa}^{\mathsf{T}} \\ M(I + \mathscr{U}_{2\mathrm{b},\kappa+1}\mathscr{U}_{3\mathrm{b},\kappa+1})^{-1}\mathscr{U}_{2\mathrm{b},\kappa+1}\mathscr{V}_{3\mathrm{b},\kappa+1}^{\mathsf{T}} & -M(I + \mathscr{U}_{2\mathrm{b},\kappa+1}\mathscr{U}_{3\mathrm{b},\kappa+1})^{-1}\mathscr{V}_{2\mathrm{b},\kappa+1}^{\mathsf{T}} \end{bmatrix} \begin{bmatrix} x_N \\ \lambda_N \end{bmatrix} \tag{4.36}$$

where λ_N by (4.30).

Proof. The optimal solutions can be generated via the double generating functions constructed by $F_{3\mathrm{f}}(\lambda_k, x_0, k)$ and $F_{3\mathrm{b}}(\lambda_k, x_N, k)$. We solve λ_k^* and x_k^* from (2.102) and (2.114), and substitute the expression of λ_k^* (changing the indices from k to $k+1$) into (4.6) to obtain u_k^* in (4.31) for Problem 4.1. We can also derive the optimal solutions in (4.32)–(4.36) based on the other five double generating functions by the similar way. □

It is clear that each one of the generators (4.31)–(4.36) is constructed by using only two different generating functions (double generating functions), in which the first four are based on double generating functions with opposite time directions, i.e. forward and backward, while the last two are based on the same time directions. Moreover, all of these six generators are in terms of the generating function coefficients and boundary conditions of the state. Due to such structures, we can divide the whole computation into two parts, off-line and on-line parts. In the off-line part, we calculate the generating function coefficients in advance. Then in the on-line part, we can efficiently generate optimal solutions when there comes the computational demand for different boundary conditions. From this viewpoint, the developed double generating functions method is useful for on-line repetitive computation for different boundary conditions. Among six generators, (4.31) is the most convenient one since it does not need extra computation for λ_0 or λ_N. Hence based on (4.31), we give the following algorithm to clearly show how to generate optimal solutions for different boundary conditions via double generating functions.

$\mathscr{U}_{3\mathrm{f},0} \leftarrow 0$; $\mathscr{V}_{3\mathrm{f},0} \leftarrow -I$; $\mathscr{U}_{3\mathrm{b},0} \leftarrow 0$; $\mathscr{V}_{3\mathrm{b},0} \leftarrow -I$; /* set boundary conditions */
for $k = 0, 1, \cdots, N-1$ **do**
  $\mathscr{U}_{3\mathrm{f},k+1} \leftarrow A(I + \mathscr{U}_{3\mathrm{f},k}Q)^{-1}\mathscr{U}_{3\mathrm{f},k}A^{\mathsf{T}} + G$; /* forward coefficients */
  $\mathscr{V}_{3\mathrm{f},k+1} \leftarrow \mathscr{V}_{3\mathrm{f},k}(I + Q\mathscr{U}_{3\mathrm{f},k})^{-1}A^{\mathsf{T}}$;
end
for $\kappa = 0, -1, \cdots, -N+1$ **do**
  $\mathscr{U}_{3\mathrm{b},\kappa-1} \leftarrow \mathcal{A}(I + \mathscr{U}_{3\mathrm{b},\kappa}\mathcal{Q})^{-1}\mathscr{U}_{3\mathrm{b},\kappa}\mathcal{A}^{\mathsf{T}} + \mathcal{G}$; /* backward coefficients */
  $\mathscr{V}_{3\mathrm{b},\kappa-1} \leftarrow \mathscr{V}_{3\mathrm{b},\kappa}(I + \mathcal{Q}\mathscr{U}_{3\mathrm{b},\kappa})^{-1}\mathcal{A}^{\mathsf{T}}$;
end

Algorithm 4.1: Off-line part, calculate generating function coefficients.

if *there is a computational demand for boundary conditions* $(x_{\mathrm{init}}, x_{\mathrm{term}})$ **then**
  $x_0 \leftarrow x_{\mathrm{init}}$; $x_{\mathrm{f}} \leftarrow x_{\mathrm{term}}$; /* set boundary conditions */
  for $k = 0, 1, \cdots, N-1$ **do**
    $x_k^* \leftarrow \mathscr{U}_{3\mathrm{b},\kappa}(\mathscr{U}_{3\mathrm{f},k} - \mathscr{U}_{3\mathrm{b},\kappa})^{-1}\mathscr{V}_{3\mathrm{f},k}^{\mathsf{T}}x_0 - \mathscr{U}_{3\mathrm{f},k}(\mathscr{U}_{3\mathrm{f},k} - \mathscr{U}_{3\mathrm{b},\kappa})^{-1}\mathscr{V}_{3\mathrm{b},\kappa}^{\mathsf{T}}x_N$;
    $u_k^* \leftarrow M(\mathscr{U}_{3\mathrm{f},k+1} - \mathscr{U}_{3\mathrm{b},\kappa+1})^{-1}\mathscr{V}_{3\mathrm{f},k+1}^{\mathsf{T}}x_0 - M(\mathscr{U}_{3\mathrm{f},k+1} - \mathscr{U}_{3\mathrm{b},\kappa+1})^{-1}\mathscr{V}_{3\mathrm{b},\kappa+1}^{\mathsf{T}}x_N$;
  end
else
  goto 1; /* on-demand */
end

Algorithm 4.2: On-line part, generate optimal solutions.

4.3 Invertibility Analysis

Notice the terms to be inverted (inverse terms) in generators (4.31)–(4.36), including (4.27)–(4.30). After we develop these optimal generators, the invertibility analysis is another important issue that we should pay attention to. In this section, we first show three kinds of properties of the twelve generating function coefficients in Section 4.3.1. Then, based on this, we give the conclusion of the invertibility in Section 4.3.2.

4.3.1 Properties of Generating Function Coefficients

We first show the general properties of the twelve generating function coefficients.

Lemma 4.1. *For Problem 4.1, the twelve generating function coefficients of $F_{2\mathrm{b}}$, $F_{3\mathrm{b}}$, $F_{2\mathrm{f}}$, and $F_{3\mathrm{f}}$ satisfy the following relations.*

(i) *Coefficients of $F_{2\mathrm{b}}$: $\mathscr{U}_{2\mathrm{b},\kappa} \succcurlyeq 0$; $\mathscr{V}_{2\mathrm{b},\kappa}$ is invertible; $\mathscr{W}_{2\mathrm{b},\kappa} \preccurlyeq 0$.*

(ii) *Coefficients of $F_{3\mathrm{b}}$: $\mathscr{U}_{3\mathrm{b},\kappa} \preccurlyeq 0$; $\mathscr{V}_{3\mathrm{b},\kappa}$ is invertible; $\mathscr{W}_{3\mathrm{b},\kappa} \succcurlyeq 0$.*

(iii) *Coefficients of $F_{2\mathrm{f}}$: $\mathscr{U}_{2\mathrm{f},k} \preccurlyeq 0$; $\mathscr{V}_{2\mathrm{f},k}$ is invertible; $\mathscr{W}_{2\mathrm{f},k} \succcurlyeq 0$.*

(iv) *Coefficients of $F_{3\mathrm{f}}$: $\mathscr{U}_{3\mathrm{f},k} \succcurlyeq 0$; $\mathscr{V}_{3\mathrm{f},k}$ is invertible; $\mathscr{W}_{3\mathrm{f},k} \preccurlyeq 0$.*

Proof. (i) First, we will use mathematical induction to prove $\mathscr{U}_{2\mathrm{b},\kappa} \succcurlyeq 0$. Before the induction, we rewrite (4.20) as

$$\mathscr{U}_{2\mathrm{b},\kappa-1} = A^{\mathsf{T}}(I + \mathscr{U}_{2\mathrm{b},\kappa}G)^{-1}(\mathscr{U}_{2\mathrm{b},\kappa} + \mathscr{U}_{2\mathrm{b},\kappa}G\mathscr{U}_{2\mathrm{b},\kappa})(I + G\mathscr{U}_{2\mathrm{b},\kappa})^{-1}A + Q. \tag{4.37}$$

Since $\mathscr{U}_{2\mathrm{b},0} = 0$ (hence $I + G\mathscr{U}_{2\mathrm{b},0}$ is invertible), we have $\mathscr{U}_{2\mathrm{b},-1} = Q \succcurlyeq 0$. Now, we suppose the general case that $\mathscr{U}_{2\mathrm{b},\kappa} \succcurlyeq 0$ (hence $I + G\mathscr{U}_{2\mathrm{b},\kappa}$ is invertible). It is clear that $A^{\mathsf{T}}(I + \mathscr{U}_{2\mathrm{b},\kappa}G)^{-1}(\mathscr{U}_{2\mathrm{b},\kappa} + \mathscr{U}_{2\mathrm{b},\kappa}G\mathscr{U}_{2\mathrm{b},\kappa})(I + G\mathscr{U}_{2\mathrm{b},\kappa})^{-1}A \succcurlyeq 0$ such that $\mathscr{U}_{2\mathrm{b},\kappa-1} \succcurlyeq 0$ (hence $I + G\mathscr{U}_{2\mathrm{b},\kappa-1}$ is invertible) due to (4.37). Therefore, we have the conclusion $\mathscr{U}_{2\mathrm{b},\kappa} \succcurlyeq 0$. Meanwhile, it is also guaranteed by the above induction that $I + \mathscr{U}_{2\mathrm{b},\kappa}G$ and $I + G\mathscr{U}_{2\mathrm{b},\kappa}$ are invertible.

Second, since both $I + \mathscr{U}_{2\mathrm{b},\kappa}G$ and A are invertible, then $\mathscr{V}_{2\mathrm{b},\kappa}$ is invertible according to the recurrence relation (4.21).

Third, the general property of $\mathscr{W}_{2\mathrm{b},\kappa}$ can be proven by the similar way to $\mathscr{U}_{2\mathrm{b},\kappa}$ by rewriting (4.22) as

$$(-\mathscr{W}_{2\mathrm{b},\kappa-1}) = (-\mathscr{W}_{2\mathrm{b},\kappa}) + \mathscr{V}_{2\mathrm{b},\kappa}(I + G\mathscr{U}_{2\mathrm{b},\kappa})^{-1}(G + G\mathscr{U}_{2\mathrm{b},\kappa}G)(I + \mathscr{U}_{2\mathrm{b},\kappa}G)^{-1}\mathscr{V}_{2\mathrm{b},\kappa}^{\mathsf{T}}$$

(ii)–(iv) Proofs for (ii)–(iv) are similar to the proof for (i), so they are omitted here. □

Next, we give the following lemma to show the rank properties of the generating function coefficients. In this lemma, we will use $\mathscr{U}_{\mathrm{f},k}$ as the generalization of $\mathscr{U}_{2\mathrm{f},k}$ and $\mathscr{U}_{3\mathrm{f},k}$, and $\mathscr{W}_{\mathrm{f},k}$ as the generalization of $\mathscr{W}_{2\mathrm{f},k}$ and $\mathscr{W}_{3\mathrm{f},k}$. Dually, $\mathscr{U}_{\mathrm{b},\kappa}$ and $\mathscr{W}_{\mathrm{b},\kappa}$ will be used for backward generating function coefficients.

Lemma 4.2. *For Problem 4.1, the twelve generating function coefficients of $F_{2\mathrm{b}}$, $F_{3\mathrm{b}}$, $F_{2\mathrm{f}}$, and $F_{3\mathrm{f}}$ satisfy the following relations.*

(i) $\mathrm{rank}(\mathscr{U}_{\mathrm{b},\kappa-1}) > \mathrm{rank}(\mathscr{U}_{\mathrm{b},\kappa})$ *when* $\mathscr{U}_{\mathrm{b},\kappa}$ *has deficient rank,* $\mathrm{rank}(\mathscr{U}_{\mathrm{b},\kappa-1}) = \mathrm{rank}(\mathscr{U}_{\mathrm{b},\kappa})$ *when* $\mathscr{U}_{\mathrm{b},\kappa}$ *has full rank.*

(ii) $\mathrm{rank}(\mathscr{W}_{\mathrm{b},\kappa-1}) > \mathrm{rank}(\mathscr{W}_{\mathrm{b},\kappa})$ *when* $\mathscr{W}_{\mathrm{b},\kappa}$ *has deficient rank,* $\mathrm{rank}(\mathscr{W}_{\mathrm{b},\kappa-1}) = \mathrm{rank}(\mathscr{W}_{\mathrm{b},\kappa})$ *when* $\mathscr{W}_{\mathrm{b},\kappa}$ *has full rank.*

(iii) $\mathrm{rank}(\mathscr{U}_{\mathrm{f},k+1}) > \mathrm{rank}(\mathscr{U}_{\mathrm{f},k})$ *when* $\mathscr{U}_{\mathrm{f},k}$ *has deficient rank,* $\mathrm{rank}(\mathscr{U}_{\mathrm{f},k+1}) = \mathrm{rank}(\mathscr{U}_{\mathrm{f},k})$ *when* $\mathscr{U}_{\mathrm{f},k}$ *has full rank.*

(iv) $\mathrm{rank}(\mathscr{W}_{\mathrm{f},k+1}) > \mathrm{rank}(\mathscr{W}_{\mathrm{f},k})$ *when* $\mathscr{W}_{\mathrm{f},k}$ *has deficient rank,* $\mathrm{rank}(\mathscr{W}_{\mathrm{f},k+1}) = \mathrm{rank}(\mathscr{W}_{\mathrm{f},k})$ *when* $\mathscr{W}_{\mathrm{f},k}$ *has full rank.*

Proof. (i) We take $\mathscr{U}_{2\mathrm{b},\kappa}$ as example to prove this part. To well state the proof, we write (4.20) here again

$$\mathscr{U}_{2\mathrm{b},\kappa-1} = A^{\mathsf{T}}(I + \mathscr{U}_{2\mathrm{b},\kappa}G)^{-1}\mathscr{U}_{2\mathrm{b},\kappa}A + Q.$$

The second result, $\mathrm{rank}(\mathscr{U}_{2\mathrm{b},\kappa-1}) = \mathrm{rank}(\mathscr{U}_{2\mathrm{b},\kappa})$ when $\mathscr{U}_{2\mathrm{b},\kappa}$ has full rank, is obvious since then $\mathrm{rank}(\mathscr{U}_{2\mathrm{b},\kappa-1}) = \mathrm{rank}(\mathscr{U}_{2\mathrm{b},\kappa}) = n$ according to (4.37). Thus, we mainly investigate the first result. Due to (4.37), it is clear that $\ker(\mathscr{U}_{2\mathrm{b},\kappa-1}) \subseteq \ker(Q)$ since both the summands in the right hand side of (4.37) are positive semi-definite for $\mathscr{U}_{2\mathrm{b},\kappa-1}$. Then to prove the first result, we use contradiction, $\mathrm{rank}(\mathscr{U}_{2\mathrm{b},\kappa-1}) \leqslant \mathrm{rank}(\mathscr{U}_{2\mathrm{b},\kappa})$. Since $\mathscr{U}_{2\mathrm{b},\kappa-1} - \mathscr{U}_{2\mathrm{b},\kappa} \succcurlyeq 0$ [68, 69], it can only be $\mathrm{rank}(\mathscr{U}_{2\mathrm{b},\kappa-1}) = \mathrm{rank}(\mathscr{U}_{2\mathrm{b},\kappa})$. Hence $\ker(\mathscr{U}_{2\mathrm{b},\kappa-1}) = \ker(\mathscr{U}_{2\mathrm{b},\kappa})$ such that $\ker(\mathscr{U}_{2\mathrm{b},\kappa}) \subseteq \ker(Q)$. Therefore, there exists nontrivial $y \in \mathbb{R}^n$ such that $\mathscr{U}_{2\mathrm{b},\kappa}y = 0 \Rightarrow Qy = 0 \Rightarrow A^{\mathsf{T}}(I + \mathscr{U}_{2\mathrm{b},\kappa}G)^{-1}\mathscr{U}_{2\mathrm{b},\kappa}Ay = 0 \Rightarrow \mathscr{U}_{2\mathrm{b},\kappa}Ay = 0$ due to (4.20). Apply the same argument on Ay instead of y, we have $\mathscr{U}_{2\mathrm{b},\kappa}Ay = 0 \Rightarrow QAy = 0 \Rightarrow \mathscr{U}_{2\mathrm{b},\kappa}A^2y = 0$. Continue in this manner, we get $Qy = QAy = QA^2y = \cdots = QA^{n-1}y = 0$. Then, by taking transposes, we have $y^{\mathsf{T}}[Q \;\; A^{\mathsf{T}}Q \;\; (A^{\mathsf{T}})^2Q \;\cdots\; (A^{\mathsf{T}})^{n-1}Q] = 0$ for some nonzero y, contradicting that (Q, A) is observable. Therefore, $\mathrm{rank}(\mathscr{U}_{2\mathrm{b},\kappa-1}) > \mathrm{rank}(\mathscr{U}_{2\mathrm{b},\kappa})$ when $\mathscr{U}_{2\mathrm{b},\kappa}$ has deficient rank.

This part of result can also be proven if we take $\mathscr{U}_{3\mathrm{b},\kappa}$ as example.

(ii) At the beginning, we also write (4.22) here

$$(-\mathscr{W}_{2\mathrm{b},\kappa-1}) = (-\mathscr{W}_{2\mathrm{b},\kappa}) + \mathscr{V}_{2\mathrm{b},\kappa}(I + G\mathscr{U}_{2\mathrm{b},\kappa})^{-1}G\mathscr{V}_{2\mathrm{b},\kappa}^{\mathsf{T}}.$$

First, let us consider the case when $\kappa = -1$. We use contradiction, $\mathrm{rank}(-\mathscr{W}_{2\mathrm{b},-2}) \leqslant \mathrm{rank}(-\mathscr{W}_{2\mathrm{b},-1})$. It is clear that it can only be $\mathrm{rank}(-\mathscr{W}_{2\mathrm{b},-2}) = \mathrm{rank}(-\mathscr{W}_{2\mathrm{b},-1})$ which leads to $\ker(-\mathscr{W}_{2\mathrm{b},-2}) = \ker(-\mathscr{W}_{2\mathrm{b},-1}) = \ker(G)$ due to the above expression. Thus, $\exists y' \neq 0$ such that $(-\mathscr{W}_{2\mathrm{b},-2})y' = 0 \Rightarrow Gy' = 0 \Rightarrow A(I + GQ)^{-1}GA^{\mathsf{T}}y' = 0 \Rightarrow GA^{\mathsf{T}}y' = 0$ due to (4.22) when $\kappa = -1$. Hence $\ker(G)$ is A^{T}-invariant, i.e. $G(A^{\mathsf{T}})^p y' = 0$ ($\forall p > 0$ and $\forall y' \in \ker(G)$), such that $y'^{\mathsf{T}}[G \;\; AG \;\; A^2G \;\cdots\; A^{n-1}G] = 0$ for some nonzero y', contradicting that (A, G) is controllable. Therefore, $\mathrm{rank}(-\mathscr{W}_{2\mathrm{b},-2}) > \mathrm{rank}(-\mathscr{W}_{2\mathrm{b},-1})$.

Next, let us consider the general case, i.e. whether $\mathrm{rank}(-\mathscr{W}_{2\mathrm{b},\kappa-1}) > \mathrm{rank}(-\mathscr{W}_{2\mathrm{b},\kappa})$ if $\mathrm{rank}(-\mathscr{W}_{2\mathrm{b},\kappa}) > \mathrm{rank}(-\mathscr{W}_{2\mathrm{b},\kappa+1})$. Similarly, we use contradiction and know that $\exists y'' \neq 0$ such that $(-\mathscr{W}_{2\mathrm{b},\kappa-1})y'' = 0 \Rightarrow (-\mathscr{W}_{2\mathrm{b},\kappa})y'' = 0 \Rightarrow G\mathscr{V}_{2\mathrm{b},\kappa}^{\mathsf{T}}y'' = 0$ due to (4.22). Meanwhile, $(-\mathscr{W}_{2\mathrm{b},\kappa})y'' = 0 \Rightarrow (-\mathscr{W}_{2\mathrm{b},\kappa+1})y'' = 0 \Rightarrow G\mathscr{V}_{2\mathrm{b},\kappa+1}^{\mathsf{T}}y'' = 0$ due to the condition $\mathrm{rank}(-\mathscr{W}_{2\mathrm{b},\kappa}) > \mathrm{rank}(-\mathscr{W}_{2\mathrm{b},\kappa+1})$. Further by (4.21) and Matrix Inversion Lemma, we have

$$G\mathscr{V}_{2\mathrm{b},\kappa}^{\mathsf{T}}y'' = 0 \Rightarrow GA^{\mathsf{T}}(I + \mathscr{U}_{2\mathrm{b},\kappa+1}G)^{-1}\mathscr{V}_{2\mathrm{b},\kappa+1}^{\mathsf{T}}y'' = 0$$

$$\Rightarrow GA^{\mathsf{T}}\mathscr{V}_{2\mathrm{b},\kappa+1}^{\mathsf{T}}y'' - GA^{\mathsf{T}}\mathscr{U}_{2\mathrm{b},\kappa+1}(I + G\mathscr{U}_{2\mathrm{b},\kappa+1})^{-1}G\mathscr{V}_{2\mathrm{b},\kappa+1}^{\mathsf{T}}y'' = 0$$
$$\Rightarrow GA^{\mathsf{T}}\mathscr{V}_{2\mathrm{b},\kappa+1}^{\mathsf{T}}y'' = 0.$$

Similarly, continuing in this manner leads to the contradiction to the controllability of (A, G). Thus, $\mathrm{rank}(-\mathscr{W}_{2\mathrm{b},\kappa-1}) > \mathrm{rank}(-\mathscr{W}_{2\mathrm{b},\kappa})$.

Note that the dimension of $\ker(-\mathscr{W}_{2\mathrm{b},\kappa})$ decreases when κ decreases, e.g. the number of y'' is less than y' if $\kappa < -1$. According to the above, once y'' can only be zero for the particular κ, i.e. $\mathrm{rank}(-\mathscr{W}_{2\mathrm{b},\kappa}) = n$, then $\mathrm{rank}(-\mathscr{W}_{2\mathrm{b},\kappa-1}) = n$.

This part of result can also be proven if we take $\mathscr{W}_{3\mathrm{b},\kappa}$ as example.

(iii)–(iv) Proofs for (iii)–(iv) are similar to the proofs for (i)–(ii), respectively, so they are omitted here. □

At last, we give the following lemma to show the convergence properties of the twelve generating function coefficients.

Lemma 4.3. *For Problem 4.1, the twelve generating function coefficients of $F_{2\mathrm{b}}$, $F_{3\mathrm{b}}$, $F_{2\mathrm{f}}$, and $F_{3\mathrm{f}}$ satisfy the following relations.*

(i) *When $\kappa \to -\infty$, coefficients of $F_{2\mathrm{b}}$: $\mathscr{U}_{2\mathrm{b},\kappa} \to \hat{\mathscr{U}}_{2\mathrm{b}} \succ 0$; $\mathscr{V}_{2\mathrm{b},\kappa}$ asymptotically converges to 0; $\mathscr{W}_{2\mathrm{b},\kappa} \to \hat{\mathscr{W}}_{2\mathrm{b}} \prec 0$.*

(ii) *When $\kappa \to -\infty$, coefficients of $F_{3\mathrm{b}}$: $\mathscr{U}_{3\mathrm{b},\kappa} \to \hat{\mathscr{U}}_{3\mathrm{b}} \prec 0$; $\mathscr{V}_{3\mathrm{b},\kappa}$ asymptotically converges to 0; $\mathscr{W}_{3\mathrm{b},\kappa} \to \hat{\mathscr{W}}_{3\mathrm{b}} \succ 0$.*

(iii) *When $k \to \infty$, coefficients of $F_{2\mathrm{f}}$: $\mathscr{U}_{2\mathrm{f},k} \to \hat{\mathscr{U}}_{2\mathrm{f}}^{+} \prec 0$; $\mathscr{V}_{2\mathrm{f},k}$ asymptotically converges to 0; $\mathscr{W}_{2\mathrm{f},k} \to \hat{\mathscr{W}}_{2\mathrm{f}} \succ 0$.*

(iv) *When $k \to \infty$, coefficients of $F_{3\mathrm{f}}$: $\mathscr{U}_{3\mathrm{f},k} \to \hat{\mathscr{U}}_{3\mathrm{f}} \succ 0$; $\mathscr{V}_{3\mathrm{f},k}$ asymptotically converges to 0; $\mathscr{W}_{3\mathrm{f},k} \to \hat{\mathscr{W}}_{3\mathrm{f}} \prec 0$.*

For the proof see Appendix of this chapter.

The three lemmas in this subsection show the comprehensive behaviour of the twelve generating function coefficients. They are the bases for the next invertibility analysis.

4.3.2 Invertibility Analysis

First, we present a remark.

Remark 4.2. Recall $F_{2\mathrm{f}}$ and $F_{2\mathrm{b}}$ in Proposition 4.1. We can also obtain the expressions of these two kinds of generating functions by Legendre transformations [50]. For example, we can get the expression of $F_{2\mathrm{f}}$ by

$$F_{2\mathrm{f}}(x_k, \lambda_0, k) = F_{3\mathrm{f}}(\lambda_k, x_0, k) + \lambda_0^{\mathsf{T}}x_0 + \lambda_k^{\mathsf{T}}x_k$$

through which $F_{2\mathrm{f}}$ and $F_{3\mathrm{f}}$ are related. Based on this, coefficients of $F_{2\mathrm{f}}$ can be expressed by the coefficients of $F_{3\mathrm{f}}$, e.g. the first coefficient $\mathscr{U}_{2\mathrm{f},k} = (\mathscr{V}_{3\mathrm{f},k}^{\mathsf{T}}\mathscr{W}_{3\mathrm{f},k}^{-1}\mathscr{V}_{3\mathrm{f},k} - \mathscr{U}_{3\mathrm{f},k})^{-1}$. By a similar way, we can also have $\mathscr{U}_{2\mathrm{b},\kappa} = (\mathscr{V}_{3\mathrm{b},\kappa}^{\mathsf{T}}\mathscr{W}_{3\mathrm{b},\kappa}^{-1}\mathscr{V}_{3\mathrm{b},\kappa} - \mathscr{U}_{3\mathrm{b},\kappa})^{-1}$. Note that the inverse terms here are nonsingular when k is large enough and κ is small enough, respectively.

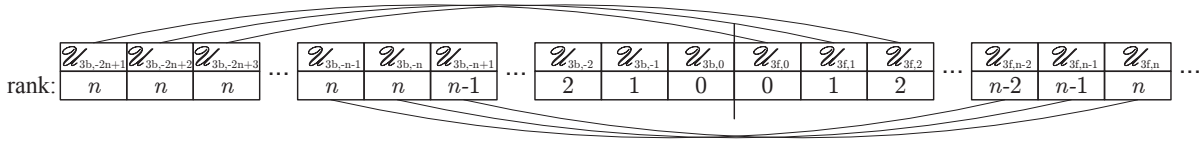


Figure 4.1: Interpretation of Theorem 4.2(v). When the increments of $\mathrm{rank}(\mathscr{U}_{3\mathrm{f},k})$ and $\mathrm{rank}(\mathscr{U}_{3\mathrm{b},\kappa})$ are both one (extreme case), the difference of each linked $\mathscr{U}_{3\mathrm{f},k}$ and $\mathscr{U}_{3\mathrm{b},k-N}$ (i.e. $\mathscr{U}_{3\mathrm{f},k} - \mathscr{U}_{3\mathrm{b},k-N}$, $k = 0, 1, \cdots, N$) is invertible, under the critical condition $N = 2n - 1$.

Based on Lemmas 4.1–4.3, and Remark 4.2, we show the invertibility of inverse terms in (4.27)–(4.36) by the following theorem.

Theorem 4.2. *For Problem 4.1, the twelve generating function coefficients of $F_{2\mathrm{b}}$, $F_{3\mathrm{b}}$, $F_{2\mathrm{f}}$, and $F_{3\mathrm{f}}$ satisfy the following relations.*

(i) *If $N \geqslant n$, $\mathscr{U}_{3\mathrm{b},-N}$ in (4.27) is invertible.*

(ii) *If $N \geqslant n$, $\mathscr{U}_{3\mathrm{f},N}$ in (4.28) is invertible.*

(iii) *If $N \geqslant n$, $\mathscr{W}_{2\mathrm{f},N}$ in (4.29) is invertible.*

(iv) *If $N \geqslant n$, $\mathscr{W}_{2\mathrm{b},-N}$ in (4.30) is invertible.*

(v) *If $N \geqslant 2n-1$, $\mathscr{U}_{3\mathrm{f},k} - \mathscr{U}_{3\mathrm{b},\kappa}$ in (4.31) is invertible, $\forall k = 0, 1, \cdots, N$.*

(vi) *$I + \mathscr{U}_{3\mathrm{f},k}\mathscr{U}_{2\mathrm{b},\kappa}$ in (4.32) is invertible, $\forall k = 0, 1, \cdots, N$.*

(vii) *$I + \mathscr{U}_{3\mathrm{b},\kappa}\mathscr{U}_{2\mathrm{f},k}$ in (4.33) is invertible, $\forall k = 0, 1, \cdots, N$.*

(viii) *If $N \geqslant 2n-1$, $\mathscr{U}_{2\mathrm{b},\kappa} - \mathscr{U}_{2\mathrm{f},k}$ in (4.34) is invertible, $\forall k = 0, 1, \cdots, N$.*

(ix) *When $k \to \infty$, $I + \mathscr{U}_{3\mathrm{f},k}\mathscr{U}_{2\mathrm{f},k} \to 0$ in (4.35).*

(x) *When $\kappa \to -\infty$, $I + \mathscr{U}_{3\mathrm{b},\kappa}\mathscr{U}_{2\mathrm{b},\kappa} \to 0$ in (4.36).*

Proof. (i) Due to Lemma 4.2(i), we know that $\mathrm{rank}(\mathscr{U}_{3\mathrm{b},\kappa})$ starts from zero and increases at each step in the beginning. Let us consider the extreme case that the increment of the rank is only one, then $\mathrm{rank}(\mathscr{U}_{3\mathrm{b},\kappa}) = n$ when $\kappa \leqslant -n$. Hence $\mathscr{U}_{3\mathrm{b},-N}$ in (4.27) is invertible if $-N \leqslant -n$, i.e. $N \geqslant n$.

(ii)–(iv) Proofs for (ii)–(iv) are similar to the proof for (i), so they are omitted here.

(v) Similarly, let us also consider the extreme case that the increments of $\mathrm{rank}(\mathscr{U}_{3\mathrm{f},k})$ and $\mathrm{rank}(\mathscr{U}_{3\mathrm{b},\kappa})$ are both one at the first n-steps. Since $\mathscr{U}_{3\mathrm{f},k} \succcurlyeq 0$ and $\mathscr{U}_{3\mathrm{b},\kappa} \preccurlyeq 0$ due to Lemma 4.1. The critical condition $N = 2n - 1$, as Figure 4.1 shows, guarantees that one variable, either $\mathscr{U}_{3\mathrm{f},k}$ or $\mathscr{U}_{3\mathrm{b},k-N}$ in the difference $\mathscr{U}_{3\mathrm{f},k} - \mathscr{U}_{3\mathrm{b},k-N}$ is (positive/negative) definite and the other one (positive/negative) semi-definite such that $\mathscr{U}_{3\mathrm{f},k} - \mathscr{U}_{3\mathrm{b},k-N}$ is always invertible, $\forall k = 0, 1, \cdots, N$. Hence it is clear to conclude that if $N \geqslant 2n - 1$, $\mathscr{U}_{3\mathrm{f},k} - \mathscr{U}_{3\mathrm{b},\kappa}$ in (4.31) is invertible, $\forall k = 0, 1, \cdots, N$.

(vi) Since both $\mathscr{U}_{3\mathrm{f},k}$ and $\mathscr{U}_{2\mathrm{b},\kappa}$ are either positive definite or positive semi-definite, eigenvalues of $I + \mathscr{U}_{3\mathrm{f},k}\mathscr{U}_{2\mathrm{b},\kappa}$ are always positive. Hence if $N \geqslant 2n-1$, $I + \mathscr{U}_{3\mathrm{f},k}\mathscr{U}_{2\mathrm{b},\kappa}$ in (4.32) is invertible, $\forall k = 0, 1, \cdots, N$.

(vii)–(viii) Proofs for (vii)–(viii) are similar to the proofs for (vi)–(v) respectively, so they are omitted here.

(ix) According to Lemma 4.3(iv), when $k \to \infty$, $\mathscr{U}_{2\mathrm{f},k} = -\mathscr{U}_{3\mathrm{f},k}^{-1}$, i.e. $\hat{\mathscr{U}}_{2\mathrm{f}} = -\hat{\mathscr{U}}_{3\mathrm{f}}^{-1}$. Hence in such a case, $I + \mathscr{U}_{3\mathrm{f},k}\mathscr{U}_{2\mathrm{f},k} \to 0$ in (4.35).

(x) According to Lemma 4.3(ii), when $\kappa \to -\infty$, $\mathscr{U}_{2\mathrm{b},\kappa} = -\mathscr{U}_{3\mathrm{b},\kappa}^{-1}$, i.e. $\hat{\mathscr{U}}_{2\mathrm{b}} = -\hat{\mathscr{U}}_{3\mathrm{b}}^{-1}$. Hence in such a case, $I + \mathscr{U}_{3\mathrm{b},\kappa}\mathscr{U}_{2\mathrm{b},\kappa} \to 0$ in (4.36). □

Remark 4.3. Invertibility of the inverse terms in second rows of generators (4.31)–(4.36) for optimal input are the same to the ones in first rows for optimal state as shown in Theorem 4.2(v)–(x), respectively, so the proofs are omitted.

Remark 4.4. Recall the inverse terms $I + \mathscr{U}_{3\mathrm{f},k}\mathscr{U}_{2\mathrm{f},k}$ in (4.35) and $I + \mathscr{U}_{3\mathrm{b},\kappa}\mathscr{U}_{2\mathrm{b},\kappa}$ in (4.36). Due to Theorem 4.2(ix), it is sure that $||I + \mathscr{U}_{3\mathrm{f},k}\mathscr{U}_{2\mathrm{f},k}||_{\max} < \varepsilon$ when $k > \delta$, where $\varepsilon \in \mathbb{R}$ is small enough and δ is the step bound corresponding to ε. Numerical computations usually performed on the digital computer that has smallest number threshold below which will be treated as zero. From this viewpoint, treat ε as such a threshold, then it is clear that $I + \mathscr{U}_{3\mathrm{f},k}\mathscr{U}_{2\mathrm{f},k}$ will be singular when $k > \delta$. This means such a term will be singular even within finite time steps in real computations that causes instability. The term $I + \mathscr{U}_{3\mathrm{b},\kappa}\mathscr{U}_{2\mathrm{b},\kappa}$ in (4.36) also has the same problem. Moreover, eigenvalues of the products $\mathscr{U}_{3\mathrm{f},k}\mathscr{U}_{2\mathrm{f},k}$ and $\mathscr{U}_{3\mathrm{b},\kappa}\mathscr{U}_{2\mathrm{b},\kappa}$ are both less than or equal to zero, or less than zero due to Lemma 4.1. Hence except the case in Theorem 4.2(ix) and (x), the invertibility of $I + \mathscr{U}_{3\mathrm{f},k}\mathscr{U}_{2\mathrm{f},k}$ and $I + \mathscr{U}_{3\mathrm{b},\kappa}\mathscr{U}_{2\mathrm{b},\kappa}$ are unclear.

Then based on the above theorem and remarks, it is straightforward to conclude the following corollary about the generators (4.31)–(4.36).

Corollary 4.1. *For Problem 4.1*

(i) *Generator (4.31) is well-defined if* $N \geqslant 2n-1$*;*

(ii) *Generator (4.32) with* λ_N *by (4.28) is well-defined if* $N \geqslant n$*;*

(iii) *Generator (4.33) with* λ_0 *by (4.27) is well-defined if* $N \geqslant n$*;*

(iv) *Generator (4.34) with* λ_0 *by (4.29) and* λ_N *by (4.30) is well-defined if* $N \geqslant 2n-1$*;*

(v) *Generator (4.35) with* λ_0 *by (4.29) is not well-defined;*

(vi) *Generator (4.36) with* λ_N *by (4.30) is not well-defined.*

As stated before Algorithms 4.1 and 4.2, the generator (4.31) is the most convenient for Problem 4.1. However, from the viewpoint of the condition shown in Corollary 4.1, the generators (4.32) and (4.33) are the priority. Nevertheless, the generators (4.35) and (4.36) should be avoided in practice.

4.4 Examples

In this section, we give two examples. The first one is to show the invertibility issue of the developed six generators, and the second one is to demonstrate the effectiveness of the double generating functions method for different boundary conditions by Algorithms 4.1 and 4.2.

Example 4.1. Consider Problem 4.1 with

$$A = \begin{bmatrix} 2 & 3 \\ 1 & 2 \end{bmatrix}, \quad B = \begin{bmatrix} 3 & 4 \\ -1 & 2 \end{bmatrix}, \quad Q = \begin{bmatrix} 2 & 3 \\ 3 & 6 \end{bmatrix}, \quad R = \begin{bmatrix} 1 & 2 \\ 2 & 5 \end{bmatrix}$$

and boundary conditions $x_{\text{init}} = [-3, 3]^{\mathsf{T}}$, $x_{\text{term}} = [5, 8]^{\mathsf{T}}$, where time steps $N = 12$.

To present the invertibility issue, we employ the six generators (4.31)–(4.36) to generate trajectory of the optimal state, respectively.

The results are presented in Figure 4.2. Figure 4.2(a)–(d) show that the generators (4.31)–(4.34) work well and generate the same optimal state trajectories since the contained inverse terms are invertible as claimed in Theorem 4.2(i)–(viii) such that these four generators are well-defined. Though the total time steps is only 12, instabilities already occur in Figure 4.2(e) and (f). The reason is that in numerical computations, the inverse terms in generators (4.35) and (4.36) are singular when the time k approaches N and 0, respectively, as stated in Theorem 4.2(ix)–(x) and Remark 4.4. By this example, it is clear that each one of (4.31)–(4.34) is available for application, whereas generators (4.35) and (4.36) should be avoided.

Example 4.2. Consider Problem 4.1 with

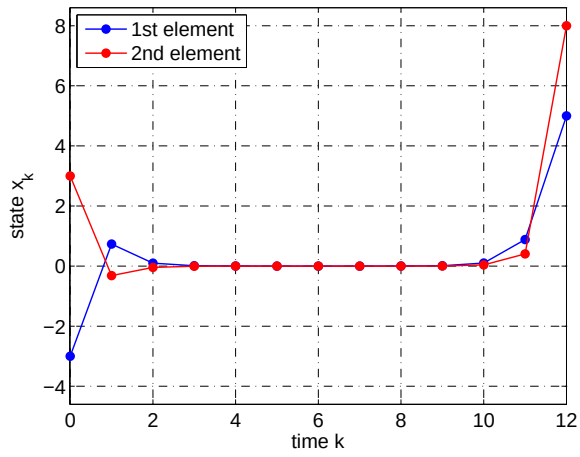
$$A = \begin{bmatrix} 3 & 1 & -1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} 1 & 1 \\ 2 & 3 \\ 1 & 2 \end{bmatrix}, \quad Q = \begin{bmatrix} 3 & 1 & 2 \\ 1 & 2 & 1 \\ 2 & 1 & 5 \end{bmatrix}, \quad R = \begin{bmatrix} 3 & 1 \\ 1 & 2 \end{bmatrix}$$

and three different sets of boundary conditions as in Table 4.1.

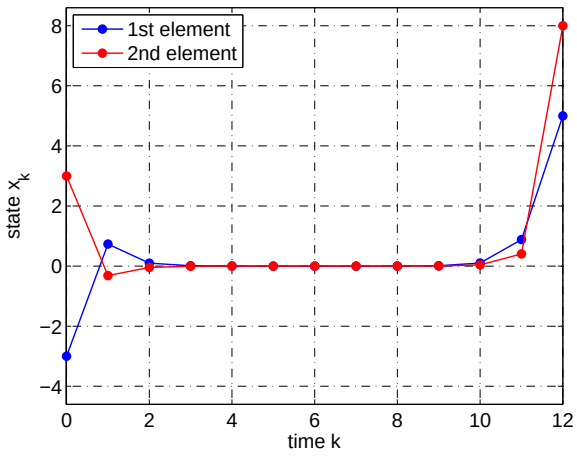
We apply Algorithms 4.1 and 4.2 for this problem. First in the off-line part, we choose the time interval as the maximum, from 0 to 20 (or larger), according to Table 4.1 to calculate the generating function coefficients. Then, optimal solutions corresponding to each particular set of boundary conditions can be efficiently generated in the on-line part. Results are presented in Figure 4.3, in which Figure 4.3(a) and (b) are the first and second elements of optimal input, and Figure 4.3(c)–(e) are the first, second, and third elements of optimal state, respectively. It can be found that trajectories of the optimal state satisfy each set of boundary conditions in Table 4.1. The off-line and on-line computational time is 0.0071 [s] and 0.0088 [s] according to Algorithms 4.1 and 4.2, respectively. The developed double generating functions method can solve such problems efficiently, especially when there is a large number of different sets of boundary conditions.

Table 4.1: Three different sets of boundary conditions.

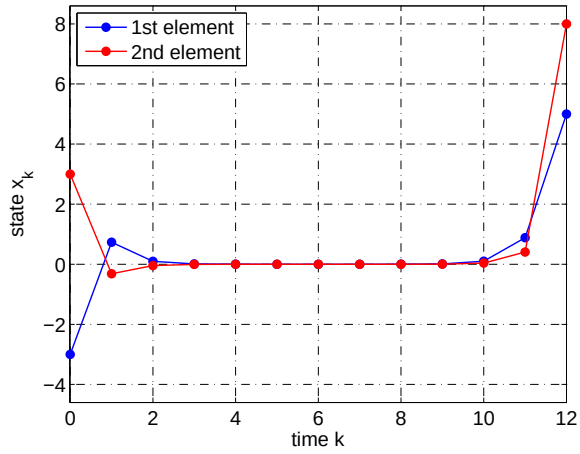
	Initial time and boundary condition	Terminal time and boundary condition
1st set	$4, [-4, 8, -2.0]^{\mathsf{T}}$	$14, [7, -7, -1.2]^{\mathsf{T}}$
2nd set	$2, [-3, 7, -1.5]^{\mathsf{T}}$	$17, [6, -6, -1.0]^{\mathsf{T}}$
3rd set	$0, [-2, 6, -1.0]^{\mathsf{T}}$	$20, [5, -5, -0.8]^{\mathsf{T}}$



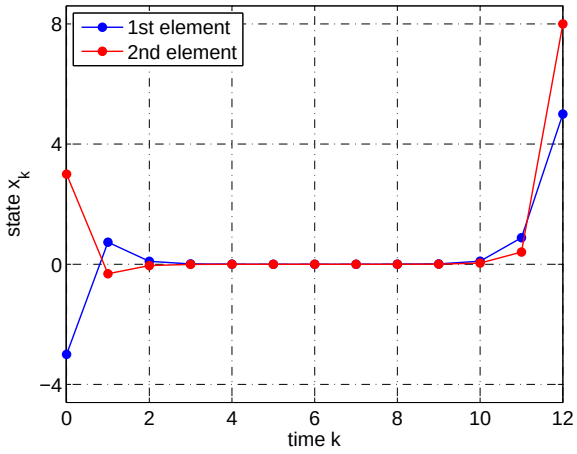
(a) Optimal state by (4.31)



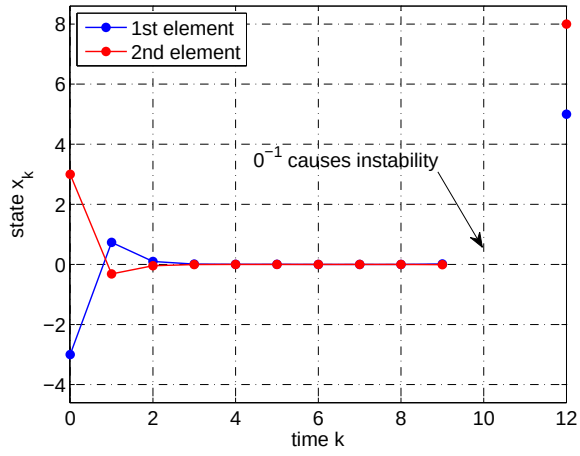
(b) Optimal state by (4.32)



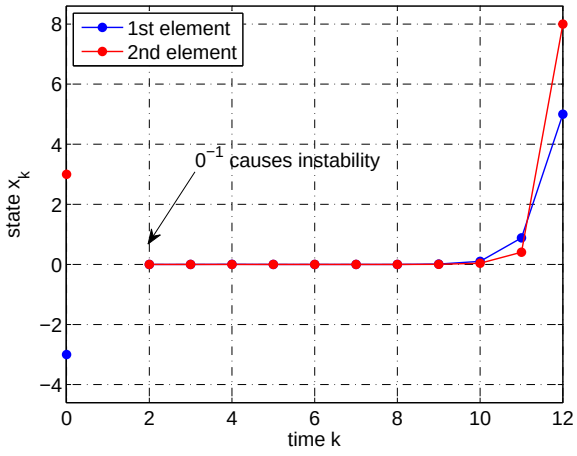
(c) Optimal state by (4.33)



(d) Optimal state by (4.34)

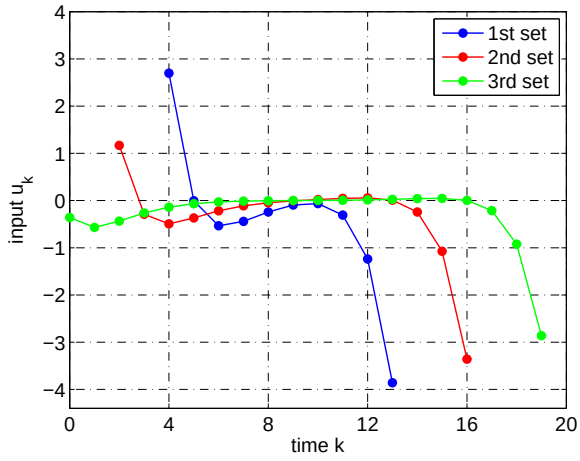


(e) Optimal state by (4.35)

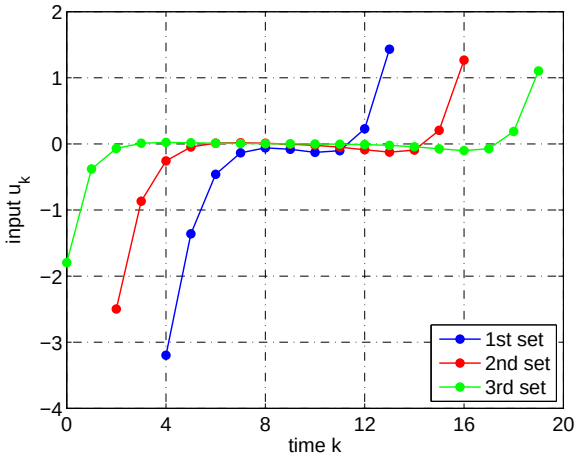


(f) Optimal state by (4.36)

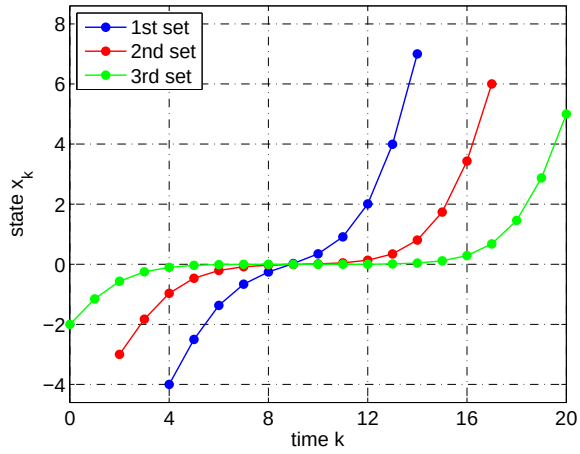
Figure 4.2: Trajectories of optimal state generated by (4.31)–(4.36).



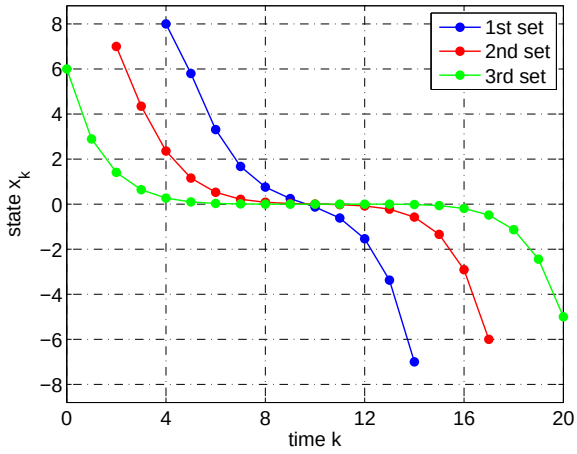
(a) Optimal input (1st element)



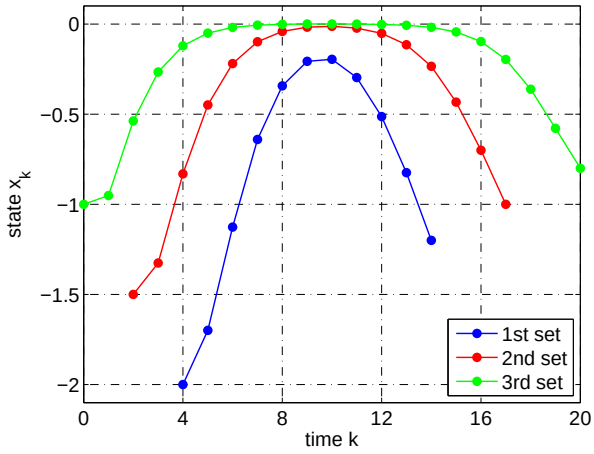
(b) Optimal input (2nd element)



(c) Optimal state (1st element)



(d) Optimal state (2nd element)



(e) Optimal state (3rd element)

Figure 4.3: Optimal input and state for the three different sets of boundary conditions.

4.5 Summary

This chapter presents a whole framework of double generating functions method to the discrete-time LQ optimal control problem, including the development of generators for optimal solutions and the numerical stability analysis. Specifically, we first derive the discrete forward and backward single generating functions by solving appropriate right and left Hamilton–Jacobi equations based on necessary conditions for optimality, and give six generators for optimal solutions based on double generating functions constructed by selecting any two different single generating functions among the candidates. Second, under the invertibility analysis of the inverse terms in these generators based on properties of the coefficients presented in this chapter, we conclude that the generators constructed by double generating functions with opposite time directions are available for applications under some mild conditions, while the generators with the same time directions should be avoided for real practice. This numerical stability analysis can also be generalized to the existing single generating function method.

Appendix

Before we prove Lemma 4.3, we first refer to the following lemma.

Lemma 4.4 ([68]). *Consider the Riccati difference equation (4.20) with the initial condition* $\mathscr{U}_{2\mathrm{b},0} = 0$. *Suppose that* (A, G) *is controllable and* (Q, A) *is observable*[†], *then*

(i) $|\sigma_i\left((I + G\mathscr{U}_{2\mathrm{b},\kappa})^{-1}A\right)| < 1$, $\forall \kappa \leqslant 0$ *and* $\forall i = 1, 2, \cdots, n$, *where* $\sigma_i(\cdot)$ *denotes the* i*-th individual eigenvalue of the matrix in the bracket.*

(ii) $\lim_{\kappa\to-\infty} \mathscr{U}_{2\mathrm{b},\kappa} = \hat{\mathscr{U}}_{2\mathrm{b}}$, *where* $\hat{\mathscr{U}}_{2\mathrm{b}}$ *is the unique stabilizing solution of the algebraic Riccati equation*

$$A^{\mathsf{T}}(I + \hat{\mathscr{U}}_{2\mathrm{b}}G)^{-1}\hat{\mathscr{U}}_{2\mathrm{b}}A - \hat{\mathscr{U}}_{2\mathrm{b}} + Q = 0.$$

Then, we give the proof for Lemma 4.3.

Proof. (i) Proof for the convergence property of $\mathscr{U}_{2\mathrm{b},\kappa}$ can be accomplished by combining Lemma 4.4(ii) with $\hat{\mathscr{U}}_{2\mathrm{b}} \succ 0$ [13].

All the eigenvalues of $(I + G\mathscr{U}_{2\mathrm{b},\kappa})^{-1}A$ have a modulus smaller than one by Lemma 4.4(i), so we know that when $\kappa \to -\infty$, $\mathscr{V}_{2\mathrm{b},\kappa}$ asymptotically converges to 0 according to (4.21).

Based on these properties, when $\kappa \to -\infty$, (4.22) becomes $\mathscr{W}_{2\mathrm{b},\kappa-1} = \mathscr{W}_{2\mathrm{b},\kappa}$ which means $\mathscr{W}_{2\mathrm{b},\kappa}$ will converge to a matrix $\hat{\mathscr{W}}_{2\mathrm{b}}$. According to Lemmas 4.1(i) and 4.2(ii), we know $\hat{\mathscr{W}}_{2\mathrm{b}} \prec 0$.

(ii) There has the corresponding lemma [68, 69] like Lemma 4.4 for the discrete Riccati equation (4.16) such that we can prove the convergence properties of the coefficients of $F_{3\mathrm{f}}$ by the same way as the above proof for (i).

(iii) For the coefficients of $F_{2\mathrm{f}}$, we can rewrite their expressions (4.12)–(4.14) as follows

$$(-\mathscr{U}_{2\mathrm{f},k+1}) = \mathcal{A}^{\mathsf{T}}\left(I + (-\mathscr{U}_{2\mathrm{f},k})(-\mathcal{G})\right)^{-1}(-\mathscr{U}_{2\mathrm{f},k})\mathcal{A} + (-\mathcal{Q})$$

[†]This condition is stricter than the ones in [68] and [13], but more applicable in this chapter.

$$\begin{aligned}\mathscr{V}_{2\mathrm{f},k+1} =& \mathscr{V}_{2\mathrm{f},k}\left(I+(-\mathcal{G})(-\mathscr{U}_{2\mathrm{f},k})\right)^{-1}\mathcal{A}\\ (-\mathscr{W}_{2\mathrm{f},k+1}) =& (-\mathscr{W}_{2\mathrm{f},k})-\mathscr{V}_{2\mathrm{f},k}\left(I+(-\mathcal{G}(-\mathscr{U}_{2\mathrm{f},k})\right)^{-1}(-\mathcal{G})\mathscr{V}_{2\mathrm{f},k}^{\mathsf{T}}.\end{aligned}$$

It is clear that $-\mathcal{Q}\succcurlyeq 0$ and $-\mathcal{G}\succcurlyeq 0$ by Remark 4.1. In addition, since $(\mathcal{A},\mathcal{G})$ is controllable and $(\mathcal{Q},\mathcal{A})$ is observable, by Kalman rank condition we know that $(\mathcal{A},-\mathcal{G})$ is controllable and $(-\mathcal{Q},\mathcal{A})$ is observable. Due to these techniques, by comparing the above three equations with (4.20)–(4.22), we can easily get the convergence results for $-\mathscr{U}_{2\mathrm{f},k}$, $\mathscr{V}_{2\mathrm{f},k}$, and $-\mathscr{W}_{2\mathrm{f},k}$: when $k\to\infty$, $-\mathscr{U}_{2\mathrm{f},k}\to\tilde{\mathscr{U}}_{2\mathrm{f}}^{+}\succ 0$, $\mathscr{V}_{2\mathrm{f},k}$ asymptotically converges to 0, and $-\mathscr{W}_{2\mathrm{f},k}\to\tilde{\mathscr{W}}_{2\mathrm{f}}\prec 0$. Therefore, we have: when $k\to\infty$, $\mathscr{U}_{2\mathrm{f},k}\to\hat{\mathscr{U}}_{2\mathrm{f}}^{+}\prec 0$, $\mathscr{V}_{2\mathrm{f},k}$ asymptotically converges to 0, and $\mathscr{W}_{2\mathrm{f},k}\to\hat{\mathscr{W}}_{2\mathrm{f}}\succ 0$.

(iv) We can prove the convergence properties of the coefficients of $F_{3\mathrm{b}}$ by the same idea of the above proof for (iii) by rewriting their expressions (4.24)–(4.26) as follows

$$\begin{aligned}(-\mathscr{U}_{3\mathrm{b},\kappa-1}) =& \mathcal{A}\left(I+(-\mathscr{U}_{3\mathrm{b},\kappa})(-\mathcal{Q})\right)^{-1}(-\mathscr{U}_{3\mathrm{b},\kappa})\mathcal{A}^{\mathsf{T}}+(-\mathcal{G})\\ \mathscr{V}_{3\mathrm{b},\kappa-1} =& \mathscr{V}_{3\mathrm{b},\kappa}\left(I+(-\mathcal{Q})(-\mathscr{U}_{3\mathrm{b},\kappa})\right)^{-1}\mathcal{A}^{\mathsf{T}}\\ (-\mathscr{W}_{3\mathrm{b},\kappa-1}) =& (-\mathscr{W}_{3\mathrm{b},\kappa})-\mathscr{V}_{3\mathrm{b},\kappa}\left(I+(-\mathcal{Q})(-\mathscr{U}_{3\mathrm{b},\kappa})\right)^{-1}(-\mathcal{Q})\mathscr{V}_{3\mathrm{b},\kappa}^{\mathsf{T}}\end{aligned}$$

and comparing with (4.16)–(4.18). □

Chapter 5

Discrete-time nonlinear optimal control problem

For the case of continuous-time problems, the references [36, 37] give analytically optimal solutions by the generating function. If we can solve the Hamilton–Jacobi equation for the generating function, it is easy for us to generate the analytically optimal input. Since the Hamilton–Jacobi equation is a nonlinear partial differential equation, it is difficult to find its analytic solution so that we need the numerical implementations to find its approximate solution. So far, there have had two numerical implementations utilized for such a purpose. One is the Galerkin spectral technique with Chebyshev polynomials [41], and the other the Taylor series expansion technique [35, 51, 43]. The first technique has the advantage of big region of convergence, but it also has the disadvantage that it requires the Hamiltonian for the optimal control problem has a special form and can not achieve the recursiveness of the ordinary differential equations for generating function coefficients. The second technique has the advantage of the recursive properties, but it also has the disadvantage that it is only applicable to systems that are small perturbations of a linear system, and it is inherently tied to the convergence of a power series for which it is difficult to estimate the region of convergence. There is trade off between these two techniques, so it is necessary for us to select the appropriate numerical implementation based on the comprehensive and deep evaluation of the problems.

For the research on discrete-time nonlinear optimal control via generating functions, it is still blank.

This chapter develops the generating function method for the general discrete-time nonlinear optimal control problems. First, we give the analytically optimal solutions, which is expressed as the state feedforward control in terms of the generating functions in Section 5.1. Then for the numerical implementations in Section 5.2, we systematically perform three steps to solve the Hamilton–Jacobi equation for the generating functions. In detail, we expand all the nonlinear functions in the Hamilton–Jacobi equation as Taylor series about zeros in tensor notations such that they can clearly present the detailed structure of the Hamilton–Jacobi equation later during the reduction. Based on this, we again employ the Taylor series technique to successfully replace one variable by the other two in the Hamilton–Jacobi equation to rewrite it by the addressed theorem in the chapter. Due to this step, we achieve our objective that the Hamilton–Jacobi equation is reduced to the difference equations for the generating function coefficients,

and they can be solved recursively with respect to the order of the Taylor series. The developed numerical framework can give the optimal solutions in terms of the pre-computed generating function coefficients and boundary conditions, such that we can divide the whole computation into two parts, the off-line part calculates the coefficients in advance, and the on-line part efficiently generates optimal solutions for different boundary conditions. From this viewpoint, it is useful for on-demand optimal solutions generation for different boundary conditions. This is summarized as an algorithm. Examples in Section 5.3 illustrate the effectiveness of the developed method. Section 5.4 summarizes this chapter.

5.1 Problem setting and analytical solutions

In this section, we formulate the discrete-time nonlinear optimal control problem in Section 5.1.1, and then give the analytically optimal solutions via generating functions in Section 5.1.2.

5.1.1 Problem setting

In this chapter, we study the discrete-time nonlinear optimal control problem, i.e. Problem 2.3 formulated in Section 2.2. We here present it again in the following to make this chapter self-contained and convenient for the reading.

Problem 5.1.

$$\min_{u} \sum_{k=0}^{N-1} \left(Q(x_k) + \frac{1}{2} u_k^{\mathsf{T}} R(x_k) u_k \right) \tag{5.1}$$

$$\text{s.t. } x_{k+1} = A(x_k) + B(x_k)u_k, \quad k = 0, 1, \cdots, N-1 \tag{5.2}$$

$$x_0 = x_{\text{init}}, \quad x_N = x_{\text{term}} \tag{5.3}$$

where the function $Q \succcurlyeq 0$, and the matrix $R(x_k) \succ 0, \forall x_k \in \mathbb{R}^n$. We assume that this problem is a convex problem.

For Problem 5.1, we give the following standard assumption.

Assumption 5.1. *Assume that the state $x_k = 0$ is an equilibrium state of the system (5.2) under the input $u_k = 0$, i.e. $A(0) = 0$.*

Based on Assumption 5.1, we have the following lemma.

Lemma 5.1. *Under Assumption 5.1, if $x_0 = x_N = 0$, the optimal state and input of the problem (5.1)–(5.3) are $\{x_k^*\}_{k=1}^{N-1} = 0$ and $\{u_k^*\}_{k=0}^{N-1} = 0$, respectively.*

Proof. Since the function $Q \succ 0$ and the matrix $R(x_k) \succ 0$ ($\forall x_k \in \mathbb{R}^n$), it is obvious that only when $\{x_k^*\}_{k=1}^{N-1} = 0$ and $\{u_k^*\}_{k=0}^{N-1} = 0$, the minimum value of the cost function (5.1) achieves the minimum zero. This holds under the conditions $A(0) = 0$ and $x_0 = x_N = 0$. □

Lemma 5.1 shows that zero sequences of state and input can be attained to Problem 5.1 under Assumption 5.1 and zero boundary conditions. This lemma will be used to prove Lemma 5.4 later in the chapter.

5.1.2 Analytical solutions via generating functions

In this chapter, we use the generating function $F_{2\mathrm{b}}$ to give analytically optimal solutions of Problem 5.1. This is presented in the following theorem.

Theorem 5.1. *The optimal input of Problem 5.1 is given as the state feedforward control*

$$u_k^* = -R(x_k)^{-1}B(x_k)^{\mathsf{T}}\frac{\partial F_{2\mathrm{b}}(x_{k+1},\lambda_N,k+1)}{\partial x_{k+1}}, \quad k=0,1,\cdots,N-1 \tag{5.4}$$

where λ_N is determined by solving the following equation

$$x_N = \left.\frac{\partial F_{2\mathrm{b}}(x_k,\lambda_N,k)}{\partial \lambda_N}\right|_{k=0}. \tag{5.5}$$

Proof. The expression of the optimal input (5.4) is obtained by the substitution of (2.111) (changing the indices from k to $k+1$) into the stationary condition (2.79). Furthermore, the terminal costate λ_N in (5.4) can be determined by solving the equation (2.112) under the condition $k=0$. This gives (5.5). □

Remark 5.1. Since u_k^* is given as the state feedforward control according to (5.4), then the substitution of this expression into the dynamics (5.2) gives the difference equation for the optimal state x_k^*. By the forward calculation from the given x_0, or backward calculation from x_N, we can get the optimal sequence $\{x_k^*\}_{k=0}^N$, and also the sequence $\{u_k^*\}_{k=0}^{N-1}$ by (5.4).

It is clear that (5.4) is expressed in terms of the generating function of the state boundary conditions x_0 and x_N. Since x_0 and x_N are pre-given, if we can find the explicit expression of the generating function, we easily determine the optimal input and state.

5.2 Numerical implementations

Since the exact expression of generating function in Theorem 5.1 can hardly be found, this section employs the Taylor series based numerical implementation to get the approximate generating function and optimal solutions. In detail, we first give the Taylor series generating function (solution to the Hamilton–Jacobi equation) and prove its recursive properties in Section 5.2.1. Then in Section 5.2.2, we present the numerically optimal solutions and summarize an algorithm to generate optimal solutions for different boundary conditions.

5.2.1 Taylor series solutions to Hamilton–Jacobi equation

Notice the analytical solutions derived in Theorem 5.1, the key point is to find the generating function which satisfies the Hamilton–Jacobi equation. Since the Hamilton–Jacobi equation (2.110) is a nonlinear partial differential equation, it is almost impossible to find its analytical solution so that we need the numerical implementations to find its approximate solution. Taylor series expansion is the most popular method utilized for such a purpose, so we here will also use this technique.

A systematic procedure of solving the Hamilton–Jacobi equation (2.110) for the generating function by the Taylor series expansion is exhibited below, including three steps. In step one, we expand all the nonlinear functions in the Hamilton–Jacobi equation as Taylor series in tensor notations. Further in step two, we represent the expanded Hamilton–Jacobi equation only in two variables by replacing the third one by an expression. Finally in step three, we reduce such Hamilton–Jacobi equation as a series of difference equations that can be solved recursively for the generating function coefficients.

- **Step one:**

We expand all the nonlinear functions in Hamilton–Jacobi equation (2.110), i.e. the generating function F and functions A, Q, and G, as Taylor series in their arguments about the origin. To do so, the following assumption is needed.

Assumption 5.2. *Assume that*

(i) $F_{2\mathrm{b}}(x_k, \lambda_N, k)$ *is an analytic function of* x_k *and* λ_N *in their neighborhood of the origin in* $\mathbb{R}^{2n}$*;*

(ii) $A(x_k)$, $Q(x_k)$, *and* $G(x_k)$ *are all analytic functions of* x_k *in its neighborhood of the origin in* $\mathbb{R}^n$.

Under Assumption 5.2, we expand the functions, $F_{2\mathrm{b}} \in \mathbb{R}$, $A \in \mathbb{R}^n$, $Q \in \mathbb{R}$, and $G \in \mathbb{R}^{n\times n}$, as Taylor series about zeros in tensor notations[†]

$$
\begin{aligned}
F_{2\mathrm{b}}(x_k, \lambda_N, k) &= F_{2\mathrm{b}}(x_k, \lambda_N; \mathscr{F}_{(\cdot,\cdot),k}) \\
&= \mathscr{F}_{(0,0),k} \\
&\quad + \left(\mathscr{F}^{\ell_1}_{(1,0),k} x_k^{\ell_1} + \mathscr{F}^{\ell_1}_{(1,1),k} \lambda_N^{\ell_1} \right) \\
&\quad + \left(\mathscr{F}^{\ell_1\ell_2}_{(2,0),k} x_k^{\ell_1} x_k^{\ell_2} + \mathscr{F}^{\ell_1\ell_2}_{(2,1),k} x_k^{\ell_1} \lambda_N^{\ell_2} + \mathscr{F}^{\ell_1\ell_2}_{(2,2),k} \lambda_N^{\ell_1} \lambda_N^{\ell_2} \right) \\
&\quad + \left(\mathscr{F}^{\ell_1\ell_2\ell_3}_{(3,0),k} x_k^{\ell_1} x_k^{\ell_2} x_k^{\ell_3} + \mathscr{F}^{\ell_1\ell_2\ell_3}_{(3,1),k} x_k^{\ell_1} x_k^{\ell_2} \lambda_N^{\ell_3} + \cdots \right) \\
&\quad + \cdots + \mathscr{F}^{\ell_1\ell_2\cdots\ell_i}_{(i,j),k} \underbrace{x_k^{\ell_1} \cdots x_k^{\ell_{i-j}}}_{i-j} \underbrace{\lambda_N^{\ell_{i-j+1}} \cdots \lambda_N^{\ell_i}}_{j} + \cdots
\end{aligned} \tag{5.6}
$$

$$
\begin{aligned}
\left(A(x_k) \right)^{\ell_1} &= \mathscr{A}^{\ell_1}_{(0)} + \mathscr{A}^{\ell_1\ell_2}_{(1)} x_k^{\ell_2} + \mathscr{A}^{\ell_1\ell_2\ell_3}_{(2)} x_k^{\ell_2} x_k^{\ell_3} + \mathscr{A}^{\ell_1\ell_2\ell_3\ell_4}_{(3)} x_k^{\ell_2} x_k^{\ell_3} x_k^{\ell_4} \\
&\quad + \cdots + \mathscr{A}^{\ell_1\ell_2\cdots\ell_{i+1}}_{(i)} \underbrace{x_k^{\ell_2} \cdots x_k^{\ell_{i+1}}}_{i} + \cdots, \quad \ell_1 = 1, 2, \cdots, n
\end{aligned} \tag{5.7}
$$

$$
Q(x_k) = \mathscr{Q}_{(0)} + \mathscr{Q}^{\ell_1}_{(1)} x_k^{\ell_1} + \mathscr{Q}^{\ell_1\ell_2}_{(2)} x_k^{\ell_1} x_k^{\ell_2} + \mathscr{Q}^{\ell_1\ell_2\ell_3}_{(3)} x_k^{\ell_1} x_k^{\ell_2} x_k^{\ell_3} + \cdots + \mathscr{Q}^{\ell_1\ell_2\cdots\ell_i}_{(i)} \underbrace{x_k^{\ell_1} \cdots x_k^{\ell_i}}_{i} + \cdots \tag{5.8}
$$

$$
\left(G(x_k) \right)^{\ell_1\ell_2} = \mathscr{G}^{\ell_1\ell_2}_{(0)} + \mathscr{G}^{\ell_1\ell_2\ell_3}_{(1)} x_k^{\ell_3} + \mathscr{G}^{\ell_1\ell_2\ell_3\ell_4}_{(2)} x_k^{\ell_3} x_k^{\ell_4} + \mathscr{G}^{\ell_1\ell_2\ell_3\ell_4\ell_5}_{(3)} x_k^{\ell_3} x_k^{\ell_4} x_k^{\ell_5}
$$

[†]In this chapter, only the generating function $F_{2\mathrm{b}}$ is used, so we do not add the subscript $2\mathrm{b}$ in its Taylor series coefficients for convenience.

$$+\cdots+\mathscr{G}_{(i)}^{\ell_1\ell_2\cdots\ell_{i+2}}\underbrace{x_k^{\ell_3}\cdots x_k^{\ell_{i+2}}}_{i}+\cdots, \quad \ell_1,\ell_2=1,2,\cdots,n \tag{5.9}$$

respectively, where the notations are explained as follows:

- In (5.6)–(5.9), ℓ_1, ℓ_2, $\cdots$, and ℓ_i are indices running from 1 to n, and they obey the Einstein summation convention, i.e. when an index appears twice in a single term, it implies summation of that term over all the values of the index.

- In (5.6)–(5.9), $x_k^{\ell_i}$ and $\lambda_N^{\ell_i}$ denote the ℓ_i-th elements of x_k and λ_N, respectively. Since x_k and λ_N are canonical, and their elements are perpendicular to each other, hence the corresponding covariant and contravariant vectors coincide with each other. For example, $(x_k)^{\ell_i}$ coincides with $(x_k)_{\ell_i}$. Hence for the sake of simplicity, we treat all the vectors as contravariant vectors such that the indices ℓ_1, ℓ_2, $\cdots$, and ℓ_i are all put in the superscript.

- In (5.6), $\mathscr{F}_{(i,j),k}^{\ell_1\ell_2\cdots\ell_i}$ denotes the general Taylor series coefficient of the generating function, where $\ell_1\ell_2\cdots\ell_i$ in the superscript implies it is an $(\ell_1,\ell_2,\cdots,\ell_i)$-th element of the $\underbrace{n\times n\times\cdots\times n}_{i}$ tensor $\mathscr{F}_{(i,j),k}$, moreover (i,j) in the subscript indicates that the product of this coefficient and the variables

  $$\mathscr{F}_{(i,j),k}^{\ell_1\ell_2\cdots\ell_i}\underbrace{x_k^{\ell_1}\cdots x_k^{\ell_{i-j}}}_{i-j}\underbrace{\lambda_N^{\ell_{i-j+1}}\cdots\lambda_N^{\ell_i}}_{j}$$

  composes the (i,j)-th order Taylor series term, and k in the subscript implies this coefficient is time-varying. Similarly in (5.7)–(5.9), $\mathscr{A}_{(i)}^{\ell_1\ell_2\cdots\ell_{i+1}}$, $\mathscr{Q}_{(i)}^{\ell_1\ell_2\cdots\ell_i}$, and $\mathscr{G}_{(i)}^{\ell_1\ell_2\cdots\ell_{i+2}}$ denote the general Taylor series coefficients of the functions A, Q, and G, respectively. They are the elements of the tensors $\mathscr{A}_{(i)}$, $\mathscr{Q}_{(i)}$, and $\mathscr{G}_{(i)}$, respectively, and the meanings of the subscript and superscript are similar to $\mathscr{F}_{(i,j),k}^{\ell_1\ell_2\cdots\ell_i}$.

- In order to reduce the number of summation terms, it is natural to require the tensor (coefficient) symmetries. Particularly, $\mathscr{F}_{(i,j),k}$ is required to be symmetric with respect to each pair of indices among $\ell_1,\cdots,\ell_{i-j}$, and also among $\ell_{i-j+1},\cdots,\ell_i$, the $\mathscr{A}_{(i)}$ is symmetric with respect to each pair of indices among $\ell_2,\cdots,\ell_{i+1}$, the $\mathscr{Q}_{(i)}$ is symmetric with respect to each pair of indices among $\ell_1,\cdots,\ell_i$, and the $\mathscr{G}_{(i)}$ is symmetric with respect to each pair of indices among $\ell_3,\cdots,\ell_{i+2}$.

- It should be noted that $\mathscr{F}_{(i,j),k}^{\ell_1\ell_2\cdots\ell_i}$ is undetermined, while the other three $\mathscr{A}_{(i)}^{\ell_1\ell_2\cdots\ell_{i+1}}$, $\mathscr{Q}_{(i)}^{\ell_1\ell_2\cdots\ell_i}$, and $\mathscr{G}_{(i)}^{\ell_1\ell_2\cdots\ell_{i+2}}$ are determined since the functions A, Q, and G are known functions. The objective of this subsection is to determine the unknown $\mathscr{F}_{(i,j),k}^{\ell_1\ell_2\cdots\ell_i}$'s.

Based on the above Taylor series expansions, the following lemma presents the coefficient properties of the functions A and Q, which will be used in the next two steps.

Lemma 5.2. *Under Assumptions 5.1 and 5.2(ii), the values of the zero and first order Taylor series coefficients of the functions A and Q are as follows*

(i) $\mathscr{A}_{(0)}^{\ell_1}=0, \quad \forall\ell_1=1,2,\cdots,n;$

(ii) $\mathscr{Q}_{(1)}^{\ell_1} = 0, \quad \forall \ell_1 = 1, 2, \cdots, n.$

Proof. (i) Since $A(0) = 0$ by Assumption 5.1, it is obvious that $\mathscr{A}_{(0)}^{\ell_1} = 0$, $\forall \ell_1 = 1, 2, \cdots, n$, according to Taylor series expression (5.7).

(ii) Since the function Q is positive definite, we have $Q(0) \leqslant Q(x_k)$ for all x_k in the neighborhood of the origin in $\mathbb{R}^n$. Moreover, Q is differentiable at 0 according to Assumption 5.2(ii). Therefore, we have $\mathscr{Q}_{(1)}^{\ell_1} = \frac{\partial Q}{\partial x_k}\Big|_{x_k=0} = 0$ by Fermat's theorem. □

- **Step two:**

We substitute (5.6)–(5.9) into (2.110) to get the expanded Hamilton–Jacobi equation, which is an equation in terms of three variables, x_{k-1}, x_k, and λ_N. In order to solve it, it is necessary for us to rewrite it only in terms of two variables. This can be achieved by replacing x_k with an expression in terms of x_{k-1} and λ_N. In detail, we substitute (2.111) into (2.81) (changing indices from k to $k-1$) to have

$$\left(\Phi(x_k, x_{k-1}, \lambda_N, k) \colon = x_k - \frac{\partial H^+(x_{k-1}, \lambda_k)}{\partial \lambda_k}\Big|_{\lambda_k = \frac{\partial F_{2\mathrm{b}}(x_k, \lambda_N, k)}{\partial x_k}} \right) = 0. \tag{5.10}$$

By solving (5.10) for x_k, we can get our desired expression

$$x_k = X\left(x_{k-1}, \lambda_N; \mathscr{F}_{(\cdot,\cdot),k}\right). \tag{5.11}$$

This is the objective of step two.

Since the equation (5.10) is nonlinear in x_k, we can not solve it analytically for x_k. For this reason, we again try to find the Taylor series solutions to x_k. According to (5.10), it is clear that x_k is an analytic function of x_{k-1} and λ_N in their neighborhood of the origin in $\mathbb{R}^{2n}$ because the functions $F_{2\mathrm{b}}$ and H^+ are both analytic functions. Based on this, we can expand x_k as Taylor series in x_{k-1} and λ_N about zeros in tensor notations

$$\begin{aligned} x_k^{\ell_1} =& \mathscr{X}_{(0,0),k}^{\ell_1} \\ &+ \left(\mathscr{X}_{(1,0),k}^{\ell_1\ell_2} x_{k-1}^{\ell_2} + \mathscr{X}_{(1,1),k}^{\ell_1\ell_2} \lambda_N^{\ell_2} \right) \\ &+ \left(\mathscr{X}_{(2,0),k}^{\ell_1\ell_2\ell_3} x_{k-1}^{\ell_2} x_{k-1}^{\ell_3} + \mathscr{X}_{(2,1),k}^{\ell_1\ell_2\ell_3} x_{k-1}^{\ell_2} \lambda_N^{\ell_3} + \mathscr{X}_{(2,2),k}^{\ell_1\ell_2\ell_3} \lambda_N^{\ell_2} \lambda_N^{\ell_3} \right) \\ &+ \left(\mathscr{X}_{(3,0),k}^{\ell_1\ell_2\ell_3\ell_4} x_{k-1}^{\ell_2} x_{k-1}^{\ell_3} x_{k-1}^{\ell_4} + \mathscr{X}_{(3,1),k}^{\ell_1\ell_2\ell_3\ell_4} x_{k-1}^{\ell_2} x_{k-1}^{\ell_3} \lambda_N^{\ell_4} + \cdots \right) \\ &+ \cdots + \mathscr{X}_{(i,j),k}^{\ell_1\ell_2\cdots\ell_{i+1}} \underbrace{x_{k-1}^{\ell_2} \cdots x_{k-1}^{\ell_{i-j+1}}}_{i-j} \underbrace{\lambda_N^{\ell_{i-j+2}} \cdots \lambda_N^{\ell_{i+1}}}_{j} \\ &+ \cdots, \quad \ell_1 = 1, 2, \cdots, n \end{aligned} \tag{5.12}$$

where $\mathscr{X}_{(i,j),k}^{\ell_1\ell_2\cdots\ell_{i+1}}$ denotes the general Taylor series coefficient of the function X.

Now by substituting Taylor series (5.12) and (5.6)–(5.9) into (5.10), we get a power series of the variables (x_{k-1}, λ_N) as

$$\left(\Phi\left(x_{k-1}, \lambda_N; \mathscr{X}_{(\cdot,\cdot),k}, \mathscr{F}_{(\cdot,\cdot),k}\right)\right)^{\ell_1}$$

$$= \sum_{i=0}^{\infty} \sum_{j=0}^{i} \left(\Phi \left(\mathscr{X}_{(\cdot,\cdot),k}, \mathscr{F}_{(\cdot,\cdot),k} \right) \right)_{(i,j)}^{\ell_1 \ell_2 \cdots \ell_{i+1}} \underbrace{x_{k-1}^{\ell_2} \cdots x_{k-1}^{\ell_{i-j+1}}}_{i-j} \underbrace{\lambda_N^{\ell_{i-j+2}} \cdots \lambda_N^{\ell_{i+1}}}_{j}$$

$$= 0, \quad \forall \ell_1 = 1, 2, \cdots, n \tag{5.13}$$

where $(\Phi(\cdot,\cdot))_{(i,j)}^{\ell_1 \ell_2 \cdots \ell_{i+1}}$ is the corresponding coefficient. Since x_{k-1} and λ_N are independent and can be chosen freely, then the only way to satisfy (5.13) is vanishing each coefficient, i.e.

$$\left(\Phi \left(\mathscr{X}_{(\cdot,\cdot),k}, \mathscr{F}_{(\cdot,\cdot),k} \right) \right)_{(i,j)}^{\ell_1 \ell_2 \cdots \ell_{i+1}} = 0, \quad \forall \ell_1, \ell_2, \cdots, \ell_{i+1} = 1, 2, \cdots, n, \ \forall (i,j) = (0,0), (1,0), \cdots . \tag{5.14}$$

Once we solve the above (5.14) for the coefficient $\mathscr{X}_{(\cdot,\cdot),k}$ in terms of $\mathscr{F}_{(\cdot,\cdot),k}$, we obtain the objective expression (5.11) according to (5.12). The following theorem gives the solutions of (5.14).

Theorem 5.2. *Under Assumptions 5.1 and 5.2, (5.14) uniquely determine* $\mathscr{X}_{(i,j),k}^{\ell_1 \ell_2 \cdots \ell_{i+1}}$, $\ell_1, \ell_2, \cdots, \ell_{i+1} = 1, 2, \cdots, n$ *and* $(i,j) = (0,0), (1,0), \cdots$, *which can be expressed in terms of the generating function coefficients analytically as*

(i) *when* $(i,j) = (0,0)$

$$\mathscr{X}_{(i,j),k}^{\ell_1} = 0, \quad \forall \ell_1 = 1, 2, \cdots, n, \ \forall k = 0, 1, \cdots, N$$

(ii) *when* $(i,j) = (i,0), \cdots, (i,i-1)$ *where* $i \neq 0$

$$\mathscr{X}_{(i,j),k}^{\ell_1 \ell_2 \cdots \ell_{i+1}} = \left(\Omega_1 \left(\mathscr{F}_{(1,0),k}, \mathscr{F}_{(2,0),k}, \cdots, \mathscr{F}_{(i+1,j),k} \right) \right)^{\ell_1 \ell_2 \cdots \ell_{i+1}},$$
$$\forall \ell_1, \ell_2, \cdots, \ell_{i+1} = 1, 2, \cdots, n \tag{5.15}$$

(iii) *when* $(i,j) = (i,i)$ *where* $i \neq 0$

$$\mathscr{X}_{(i,j),k}^{\ell_1 \ell_2 \cdots \ell_{i+1}} = \left(\Omega_2 \left(\mathscr{F}_{(2,0),k}, \mathscr{F}_{(2,1),k}, \cdots, \mathscr{F}_{(i+1,j),k} \right) \right)^{\ell_1 \ell_2 \cdots \ell_{i+1}},$$
$$\forall \ell_1, \ell_2, \cdots, \ell_{i+1} = 1, 2, \cdots, n \tag{5.16}$$

where Ω_1 *and* Ω_2 *denote analytic expressions.*

For the proof see Appendix of this chapter.

Note that the only difference between the last two cases in above Theorem 5.2 is the coefficient $\mathscr{F}_{(1,0),k}$ in the expressions right hand side. This is the reason why we do not combine these two cases.

In light of Theorem 5.2, we can successfully get the objective expression (5.11) by substituting (5.15) and (5.16) into (5.12). Further, by substituting (5.11) and (5.6)–(5.9) into (2.110), we get the expanded Hamilton–Jacobi equation in terms of two variables x_{k-1} and λ_N and the generating function coefficients as

$$\Gamma \left(x_{k-1}, \lambda_N; \mathscr{F}_{(\cdot,\cdot),k-1}, \mathscr{F}_{(\cdot,\cdot),k} \right) = 0 \tag{5.17}$$

where Γ takes the form of

$$
\begin{aligned}
&\Gamma\left(x_{k-1},\lambda_N;\mathscr{F}_{(\cdot,\cdot),k-1},\mathscr{F}_{(\cdot,\cdot),k}\right)\\
&=\left(\mathscr{F}_{(0,0),k-1}+\mathscr{F}^{\ell_1}_{(1,0),k-1}x^{\ell_1}_{k-1}+\mathscr{F}^{\ell_1}_{(1,1),k-1}\lambda^{\ell_1}_N+\mathscr{F}^{\ell_1\ell_2}_{(2,0),k-1}x^{\ell_1}_{k-1}x^{\ell_2}_{k-1}+\cdots\right)\\
&\quad-\left(\mathscr{Q}_{(0)}+\mathscr{Q}^{\ell_1}_{(1)}x^{\ell_1}_{k-1}+\mathscr{Q}^{\ell_1\ell_2}_{(2)}x^{\ell_1}_{k-1}x^{\ell_2}_{k-1}+\mathscr{Q}^{\ell_1\ell_2\ell_3}_{(3)}x^{\ell_1}_{k-1}x^{\ell_2}_{k-1}x^{\ell_3}_{k-1}+\cdots\right)\\
&\quad-\left(\mathscr{F}^{\ell_1}_{(1,0),k}+2\mathscr{F}^{\ell_1\ell_2}_{(2,0),k}x^{\ell_2}_{k}+\mathscr{F}^{\ell_1\ell_2}_{(2,1),k}\lambda^{\ell_2}_N+3\mathscr{F}^{\ell_1\ell_2\ell_3}_{(3,0),k}x^{\ell_2}_{k}x^{\ell_3}_{k}+\cdots\right)\\
&\quad\cdot\left(\mathscr{A}^{\ell_1}_{(0)}+\mathscr{A}^{\ell_1\ell'_2}_{(1)}x^{\ell'_2}_{k-1}+\mathscr{A}^{\ell_1\ell'_2\ell'_3}_{(2)}x^{\ell'_2}_{k-1}x^{\ell'_3}_{k-1}+\mathscr{A}^{\ell_1\ell'_2\ell'_3\ell'_4}_{(3)}x^{\ell'_2}_{k-1}x^{\ell'_3}_{k-1}x^{\ell'_4}_{k-1}+\cdots\right)\\
&\quad+\frac{1}{2}\left(\mathscr{G}^{\ell_1\ell_2}_{(0)}+\mathscr{G}^{\ell_1\ell_2\ell_3}_{(1)}x^{\ell_3}_{k-1}+\mathscr{G}^{\ell_1\ell_2\ell_3\ell_4}_{(2)}x^{\ell_3}_{k-1}x^{\ell_4}_{k-1}+\cdots\right)\\
&\quad\cdot\left(\mathscr{F}^{\ell_1}_{(1,0),k}+2\mathscr{F}^{\ell_1\ell'_2}_{(2,0),k}x^{\ell'_2}_{k}+\mathscr{F}^{\ell_1\ell'_2}_{(2,1),k}\lambda^{\ell'_2}_N++3\mathscr{F}^{\ell_1\ell'_2\ell'_3}_{(3,0),k}x^{\ell'_2}_{k}x^{\ell'_3}_{k}+\cdots\right)\\
&\quad\cdot\left(\mathscr{F}^{\ell_2}_{(1,0),k}+2\mathscr{F}^{\ell_2\ell''_3}_{(2,0),k}x^{\ell''_3}_{k}+\mathscr{F}^{\ell_2\ell''_3}_{(2,1),k}\lambda^{\ell''_3}_N+3\mathscr{F}^{\ell_2\ell''_3\ell''_4}_{(3,0),k}x^{\ell''_3}_{k}x^{\ell''_4}_{k}+\cdots\right)\\
&\quad-\left(\mathscr{F}_{(0,0),k}+\mathscr{F}^{\ell_1}_{(1,0),k}x^{\ell_1}_{k}+\mathscr{F}^{\ell_1}_{(1,1),k}\lambda^{\ell_1}_N+\mathscr{F}^{\ell_1\ell_2}_{(2,0),k}x^{\ell_1}_{k}x^{\ell_2}_{k}+\cdots\right)\\
&\quad+\left(\mathscr{F}^{\ell_1}_{(1,0),k}+2\mathscr{F}^{\ell_1\ell_2}_{(2,0),k}x^{\ell_2}_{k}+\mathscr{F}^{\ell_1\ell_2}_{(2,1),k}\lambda^{\ell_2}_N+3\mathscr{F}^{\ell_1\ell_2\ell_3}_{(3,0),k}x^{\ell_2}_{k}x^{\ell_3}_{k}+\cdots\right)\cdot x^{\ell_1}_{k}.
\end{aligned}
$$

- **Step three:**

Similar to the procedure from (5.13) to (5.14) of solving the equation (5.10), here we again write the expanded Hamilton–Jacobi equation (5.17) in power series

$$
\begin{aligned}
&\Gamma\left(x_{k-1},\lambda_N;\mathscr{F}_{(\cdot,\cdot),k-1},\mathscr{F}_{(\cdot,\cdot),k}\right)\\
&=\sum_{i=0}^{\infty}\sum_{j=0}^{i}\left(\mathscr{F}^{\ell_1\ell_2\cdots\ell_i}_{(i,j),k-1}-\left(\Gamma\left(\mathscr{F}_{(\cdot,\cdot),k}\right)\right)^{\ell_1\ell_2\cdots\ell_i}_{(i,j)}\right)\underbrace{x^{\ell_1}_{k-1}\cdots x^{\ell_{i-j}}_{k-1}}_{i-j}\underbrace{\lambda^{\ell_{i-j+1}}_N\cdots\lambda^{\ell_i}_N}_{j}\\
&=0
\end{aligned}
\tag{5.18}
$$

where $\left(\Gamma(\cdot)\right)^{\ell_1\ell_2\cdots\ell_i}_{(i,j)}$ is the corresponding coefficient. Again, since x_{k-1} and λ_N are independent and can be chosen freely, then the only way to satisfy (5.18) is vanishing each coefficient, i.e.

$$
\mathscr{F}^{\ell_1\ell_2\cdots\ell_i}_{(i,j),k-1}=\left(\Gamma\left(\mathscr{F}_{(\cdot,\cdot),k}\right)\right)^{\ell_1\ell_2\cdots\ell_i}_{(i,j)},\quad\forall\ell_1,\ell_2,\cdots,\ell_i=1,2,\cdots,n,\ \forall(i,j)=(0,0),(1,0),\cdots.
\tag{5.19}
$$

For these backward difference equations, the following lemma gives their terminal values.

Lemma 5.3. *Under Assumption 5.2(i), the terminal time values of the generating function coefficients, i.e.* $\mathscr{F}^{\ell_1\ell_2\cdots\ell_i}_{(i,j),k=N}$, *are as follows*

(i) *when* $(i,j)=(2,1)$

$$
\mathscr{F}^{\ell_1\ell_2}_{(i,j),N}=\begin{cases}1, & \ell_1=\ell_2\\ 0, & \ell_1\neq\ell_2\end{cases}
\tag{5.20}
$$

(ii) *when* $(i,j) \neq (2,1)$

$$\mathscr{F}^{\ell_1\ell_2\cdots\ell_i}_{(i,j),N} = 0, \quad \forall \ell_1, \ell_2, \cdots, \ell_i = 1, 2, \cdots, n. \tag{5.21}$$

Proof. By the relations (2.111) and (2.112), we have

$$F_{2\mathrm{b}}(x_k, \lambda_N, k)|_{k=N} = x_N^{\mathsf{T}} \lambda_N. \tag{5.22}$$

Then according to Taylor series expression (5.6), the above (5.22) leads to

$$\mathscr{F}^{\ell_1\ell_2}_{(2,1),N} = \begin{cases} 1, & \ell_1 = \ell_2 \\ 0, & \ell_1 \neq \ell_2 \end{cases}$$

which is (5.20) for Lemma 5.3(i), and

$$\mathscr{F}^{\ell_1\ell_2\cdots\ell_i}_{(i,j),N} = 0, \quad \forall \ell_1, \ell_2, \cdots, \ell_i = 1, 2, \cdots, n$$

for $(i,j) \neq (2,1)$, which is (5.21) for Lemma 5.3(ii). □

Now with the terminal values presented in Lemma 5.3, we can solve each difference equation in (5.19) step by step backward from $k = N$ to $k = 1$ to get the coefficient values the whole time steps.

For the difference equations, the recursiveness is firm important. It will be verified below that equations in (5.19) also maintain such properties, before which we first present a lemma about the coefficient $\mathscr{F}_{(1,0),k}$.

Lemma 5.4. *Under Assumptions 5.1 and 5.2, for the Taylor series in (5.6), we have*

$$\mathscr{F}^{\ell_1}_{(1,0),k} = 0, \quad \forall \ell_1 = 1, 2, \cdots, n, \ \forall k = 0, 1, \cdots, N. \tag{5.23}$$

Proof. To prove this lemma, we first prove the sequence $\{\lambda_k^*\}_{k=0}^N = 0$. According to the dynamic programming [48], we have

$$\lambda_k^* = \left.\frac{\partial V(x_k)}{\partial x_k}\right|_{x_k = x_k^*} \tag{5.24}$$

where the value function

$$V(x_k) \colon = \min_{u_i} \sum_{i=k}^{N-1} \left(Q(x_i) + \frac{1}{2} u_i^{\mathsf{T}} R(x_i) u_i \right).$$

It is obvious that $0 = V(0) \leqslant V(x_k)$ for all x_k in the neighborhood of the origin in $\mathbb{R}^n$. Then under the condition of $x_0 = x_N = 0$ ($\{x_k\}_{k=0}^N = 0$ by Lemma 5.1), we have $\lambda_k^* = \left.\frac{\partial V(x_k)}{\partial x_k}\right|_{x_k = x_k^* = 0} = 0$ by Fermat's theorem.

Second, according to (2.111) and (5.6), we have

$$(\lambda_k)^{\ell_1} = \left(\frac{\partial F_{2\mathrm{b}}(x_k, \lambda_N, k)}{\partial x_k} \right)^{\ell_1} = \mathscr{F}^{\ell_1}_{(1,0),k} + 2\mathscr{F}^{\ell_1\ell_2}_{(2,0),k} x_k^{\ell_2} + \mathscr{F}^{\ell_1\ell_2}_{(2,1),k} \lambda_N^{\ell_2} + \cdots,$$

$$\ell_1 = 1, 2, \cdots, n.$$

Then under the condition of $x_0 = x_N = 0$, substitution of $\{x_k^*\}_{k=0}^N = 0$ and $\{\lambda_k^*\}_{k=0}^N = 0$ into the above equation leads to

$$\mathscr{F}_{(1,0),k}^{\ell_1} = 0, \quad \forall \ell_1 = 1, 2, \cdots, n, \ \forall k = 0, 1, \cdots, N.$$

Further, since the generating function coefficients are independent of the state boundary conditions according to the difference equations in (5.19), the above result holds not only for the boundary condition $x_0 = x_N = 0$, but also for all other boundary conditions. □

Based on Lemma 5.4, now we show the recursiveness of the difference equations in (5.19) by the following theorem.

Theorem 5.3. *Under Assumptions 5.1 and 5.2, the difference equations in (5.19) can be solved recursively for the generating function coefficients $\mathscr{F}_{(i,j),k}$ with respect to the Taylor series order index (i, j).*

For the proof see Appendix of this chapter.

As is known, the recursiveness has many advantages. One of its most important benefits is that for the computation, we do not need iterations. Once we calculate the greater order coefficients based on the obtained $\mathscr{F}_{(\cdot,\cdot),k}$ (suppose from $(0, 0)$ to (i, j)), we do not require to iterate all the difference equations again. We only need to recall the obtained ones, and calculate the difference equations from the order $(i, j + 1)$. This will be clearly shown in the next subsection.

This subsection exhibits a systematic procedure of solving the Hamilton–Jacobi equation (2.110) for the generating function by Taylor series techniques in three steps. It is guaranteed by Theorem 5.2 and 5.3 that we can successfully solve the generating function coefficients.

5.2.2 Algorithm for numerically optimal solutions

Once we obtain the generating function coefficients, we obtain the generating function according to (5.6), which is the Taylor series solutions to Hamilton–Jacobi equation. By substituting it into (5.4), we can get the numerically optimal solutions. Notice the difference equations (5.19) presented in the above subsection, the generating function coefficients are independent on the state boundary conditions (5.3). In light of this, the developed generating function method is useful in on-demand optimal solutions generation for the same problem with different boundary conditions. This is the advantage of this developed method. In detail, we can divide the whole computation into two parts, in which the off-line part calculates the generating function coefficients in advance, while the on-line part can efficiently generate optimal solutions for different boundary conditions. This is summarized in the following algorithms.

```
1  𝒩 ← 𝒩₀;                                   /* set truncated Taylor series order */
2  for i = 0, 1, ··· , 𝒩₀ do
3      for j = 0, 1, ··· , i do
4          if (i, j) = (2, 1) then
5              for ℓ₁ = 1, 2, ··· , n do
6                  for ℓ_i = 1, 2, ··· , n do
7                      if ℓ₁ = ℓ_i then
8                          ℱ^{ℓ₁ℓ₂}_{(i,j),N} ← 1;          /* set terminal conditions */
9                      else
10                         ℱ^{ℓ₁ℓ₂}_{(i,j),N} ← 0;
11                     end
12                 end
13             end
14         else
15             for ℓ₁ = 1, 2, ··· , n do
16                 for ℓ₂ = 1, 2, ··· , n do
17                     ⋱
18                     for ℓ_i = 1, 2, ··· , n do
19                         ℱ^{ℓ₁ℓ₂···ℓ_i}_{(i,j),N} ← 0;
20                     end
21                     ⋰
22                 end
23             end
24         end
25     end
26 end
27 for i = 0, 1, ··· , 𝒩₀ do
28     for j = 0, 1, ··· , i do
29         for k = N, N − 1, ··· , 1 do
30             for ℓ₁ = 1, 2, ··· , n do
31                 for ℓ₂ = 1, 2, ··· , n do
32                     ⋱
33                     for ℓ_i = 1, 2, ··· , n do
34                         ℱ^{ℓ₁ℓ₂···ℓ_i}_{(i,j),k−1} ← (Γ(ℱ_{(·,·),k}))^{ℓ₁ℓ₂···ℓ_i}_{(i,j)};   /* coefficients */
35                     end
36                     ⋰
37                 end
38             end
39         end
40     end
41 end
```

Algorithm 5.1: Off-line part, calculate generating function coefficients.

1 **if** *there is a computational demand for boundary conditions* $(x_{\text{init}}, x_{\text{term}})$ **then**
2 $x(t_0) \leftarrow x_{\text{init}}$, $x(t_{\text{f}}) \leftarrow x_{\text{term}}$; /* set boundary conditions */
3 solve λ_N from $x_N = \left.\frac{\partial F_{2\text{b}}(x_k, \lambda_N, k)}{\partial \lambda_N}\right|_{k=0}$; /* calculate terminal costate */
4 **for** $k = 0, 1, \cdots, N-1$ **do**
5 $u_k^* \leftarrow -R(x_k)^{-1} B(x_k)^{\mathsf{T}} \frac{\partial F_{2\text{b}}(x_{k+1}, \lambda_N, k+1)}{\partial x_{k+1}}$; /* generate optimal input */
6 **end**
7 **else**
8 **goto** 1; /* on-demand */
9 **end**

Algorithm 5.2: On-line part, generate optimal solutions.

Notice the step 4 in Algorithm 5.2, according to Remark 5.1, it is free of us to perform either the forward calculation (i.e. $k = 0, 1, \cdots, N-1$) or the backward calculation (i.e. $k = N, N-1, \cdots, 1$) to solve the optimal solutions.

In Algorithm 5.1, optimal solutions will be more accurate if we select greater $\mathcal{N}$, i.e. expand functions as Taylor series up to higher orders. Since the computation of the coefficients is implemented off-line, it is free of us to choose any particular orders. From this viewpoint, though the original problem is nonlinear, we can still obtain its optimal solutions accurately by the developed method. However, when we increase the order $\mathcal{N}$, the total number of difference equations in the off-line part for the coefficients also increases. For this reason, when we select the order $\mathcal{N}$, both the demand of the accuracy and the computational ability of the off-line computer should be taken into account. On the other hand, it is also convenient for us to increase the order $\mathcal{N}$ due to the recursiveness of the off-line difference equations. For example to increase the order $\mathcal{N}_0$ to $\mathcal{N}_1$, there is no need to calculate coefficients from $i = 0$ to $\mathcal{N}_0$ again. We only need to read the obtained ones ($i = 0$ to $\mathcal{N}_0$), and calculate the difference equations from the order $(\mathcal{N}_0 + 1, 0)$ to $(\mathcal{N}_1, \mathcal{N}_1)$. This reduces the computational burden.

Besides the above issues, we also need to remark on whether it is necessary to adjust the truncated Taylor series orders according to the boundary conditions. This issue is related to the convergence of the Taylor series that as its order increases, the region of convergence also increases. Problem with different boundary conditions needs different convergence regions, i.e. different truncated Taylor series orders, so that in principle, it is necessary to adjust its orders according to the boundary conditions. However, in practice, if the distances from different boundary conditions to the origin are quite different, it is easy for us to make the adjustments on-line because we usually well prepare in the off-line part that the generating function coefficient orders are expanded as high as possible. If the distances from different boundary conditions to the origin are similar, it is obvious to use the same truncated Taylor series orders for all the boundary conditions. From this viewpoint, the generating function method is still useful and efficient in on-line repetitive solutions generation for different boundary conditions.

5.3 Examples

In this section, we give two examples. The first one is to show the procedure of generating optimal solutions and the accuracy of the numerical technique, and the second one is to demon-

strate the effectiveness of the generating function method for different boundary conditions by Algorithms 5.1 and 5.2.

Example 5.1. Consider the problem

$$\begin{aligned}
&\min_{u} \sum_{k=0}^{N-1} \frac{1}{2}\left(x_k^2 + u_k^2\right) \\
&\text{s.t. } x_{k+1} = \sin(x_k) + x_k + u_k, \quad k = 0, 1, \cdots, N-1 \\
&\qquad x_0 = -0.1, \quad x_N = -0.1
\end{aligned}$$

where the time steps $N = 10$.

To clearly show the procedure of generating optimal solutions by the developed generating function method, we set the above scalar problem. For this problem, we will generate the optimal solutions corresponding to different truncated Taylor series orders $\mathcal{N} = 2$, 3, and 4 (i.e. Taylor series truncated up to the 2nd, 3rd, and 4th orders).

In step one, we expand the nonlinear functions A, Q, and G as Taylor series up to the highest 4th order ($\mathcal{N} = 4$)

$$\begin{aligned}
A(x_k) &= 0 + 2 \cdot x_k + 0 \cdot x_k^2 - 1/6 \cdot x_k^3 + 0 \cdot x_k^4 + \cdots \\
Q(x_k) &= 0 + 0 \cdot x_k + 1/2 \cdot x_k^2 + 0 \cdot x_k^3 + 0 \cdot x_k^4 + \cdots \\
G(x_k) &= 1 + 0 \cdot x_k + 0 \cdot x_k^2 + 0 \cdot x_k^3 + 0 \cdot x_k^4 + \cdots
\end{aligned}$$

which verify Lemma 5.2. Note that though the above approximated functions truncated up to the 3rd order and the 4th order are the same, but the generated optimal solutions later will still be different because they depend on the generating function coefficients which are different. In step two, by solving (5.10) and performing the procedure from (5.13) to (5.14), we obtain the objective expression

$$x_k = \mathscr{X}_{(0,0),k} + \mathscr{X}_{(1,0),k} x_{k-1} + \mathscr{X}_{(1,1),k} \lambda_N + \cdots + \mathscr{X}_{(4,4),k} \lambda_N^4 + \cdots$$

where the coefficients $\mathscr{X}_{(\cdot,\cdot),k}$ are expressed as

$$\begin{aligned}
\mathscr{X}_{(0,0),k} &= 0 \\
\mathscr{X}_{(1,0),k} &= 2/(2\mathscr{F}_{(2,0),k} + 1) \\
\mathscr{X}_{(1,1),k} &= -\mathscr{F}_{(2,1),k}/(2\mathscr{F}_{(2,0),k} + 1) \\
\mathscr{X}_{(2,0),k} &= -12\mathscr{F}_{(3,0),k}/(2\mathscr{F}_{(2,0),k} + 1)^3 \\
\mathscr{X}_{(2,1),k} &= -4 \cdot (\mathscr{F}_{(3,1),k} + 2\mathscr{F}_{(2,0),k}\mathscr{F}_{(3,1),k} - 3\mathscr{F}_{(2,1),k}\mathscr{F}_{(3,0),k})/(2\mathscr{F}_{(2,0),k} + 1)^3 \\
&\vdots
\end{aligned}$$

which are all in terms of the generating function coefficients that verify Theorem 5.2. Based on this, in step three, we are able to solve the expanded Hamilton–Jacobi equation (5.18) to get the difference equations (5.19) for the generating function coefficients

Table 5.1: The computed sequences of optimal state x_k, $k = 0, 1, \cdots, 10$, corresponding to different truncated Taylor series orders $\mathcal{N} = 2$, 3, and 4.

	$\mathcal{N}=2$	$\mathcal{N}=3$	$\mathcal{N}=4$
x_0	-657975.1651	-0.1080	-0.0999
x_1	-125663.7044	-0.0412	-0.0382
x_2	-24001.7557	-0.0158	-0.0146
x_3	-4586.8157	-0.0061	-0.0057
x_4	-880.0941	-0.0026	-0.0024
x_5	-164.4839	-0.0017	-0.0016
x_6	-2.1448	-0.0025	-0.0024
x_7	-0.3878	-0.0057	-0.0057
x_8	-0.1069	-0.0146	-0.0146
x_9	-0.0767	-0.0382	-0.0382
x_{10}	-0.1	-0.1	-0.1

Table 5.2: The computed sequences of optimal input u_k, $k = 0, 1, \cdots, 9$, corresponding to different truncated Taylor series orders $\mathcal{N} = 2$, 3, and 4.

	$\mathcal{N}=2$	$\mathcal{N}=3$	$\mathcal{N}=4$
u_0	532319.6922	0.1745	0.1616
u_1	101672.6731	0.0666	0.0617
u_2	19429.5446	0.0254	0.0236
u_3	3727.4561	0.0097	0.0089
u_4	695.8896	0.0035	0.0032
u_5	8.9988	0.0009	0.0008
u_6	1.5275	-0.0008	-0.0008
u_7	0.2734	-0.0032	-0.0033
u_8	0.0296	-0.0089	-0.0089
u_9	-0.0236	-0.0236	-0.0236

Table 5.3: The three minimum cost function values corresponding to different truncated Taylor series orders $\mathcal{N} = 2$, 3, and 4.

	$\mathcal{N}=2$	$\mathcal{N}=3$	$\mathcal{N}=4$
Cost function value	371707040948.8657	0.0258	0.0223

$$\mathscr{F}_{(0,0),k-1} = \mathscr{F}_{(0,0),k}$$
$$\mathscr{F}_{(1,0),k-1} = 0$$

$$
\begin{aligned}
\mathscr{F}_{(1,1),k-1} &= \mathscr{F}_{(1,1),k} \\
\mathscr{F}_{(2,0),k-1} &= 5/2 - 2/(2\mathscr{F}_{(2,0),k} + 1) \\
\mathscr{F}_{(2,1),k-1} &= 2\mathscr{F}_{(2,1),k}/(2\mathscr{F}_{(2,0),k} + 1) \\
&\vdots
\end{aligned}
$$

which are recursive with respect to the Taylor series order index (i, j) that verify Theorem 5.3. Notice the second difference equation above, it verifies Lemma 5.4. Now with the terminal values in Lemma 5.3, by the backward time calculation from $k = N$ to 1 for each difference equation above sequentially, we obtain the values of the coefficients from $\mathscr{F}_{(0,0),k}$ to $\mathscr{F}_{(4,4),k}$, $k = 0, 1, \cdots, 10$. Because of the recursiveness, the obtained coefficients from $\mathscr{F}_{(0,0),k}$ to $\mathscr{F}_{(2,2),k}$ and to $\mathscr{F}_{(3,3),k}$ are available for the cases $\mathcal{N} = 2$ and 3, respectively. This is the advantage of the recursiveness. Now by substituting the generating function with these coefficients into Theorem 5.1, we obtain the optimal solutions.

The sequences of optimal state and input corresponding to different truncated Taylor series orders $\mathcal{N} = 2$, 3, and 4 are presented in Tables 5.1 and 5.2, respectively. In these two tables, the state and the associated input satisfy the dynamics $x_{k+1} = \sin(x_k) + x_k + u_k$. Moreover, from Table 5.1, we can find that the three state sequences are strictly fixed as the given value (-0.1) in the terminal, while along the backward time steps, they gradually separate, especially the sequence corresponding to $\mathcal{N} = 2$ diverges. The reason is because the truncated order $\mathcal{N} = 2$ is not high enough. Further, since we solve the equation (5.4) for the optimal solutions, we choose the backward time calculation[†]. Therefore, errors accumulate along the backward time due to the numerical computation, such that it is reasonable that the greatest error exists at the initial time in Table 5.1. It can also be found from Table 5.1 that among the three initial values, the one (-0.0999) corresponding to the truncated order 4, has the least error compared with the given initial value (-0.1). Hence, as the truncated order increases, the computed initial error decreases, which implies greater order $\mathcal{N}$ gives the higher accuracy. Moreover, we also give the minimum cost function value corresponding to these three orders in Table 5.3. It is clear that greater order $\mathcal{N}$ provides the less cost function value. Based on these results, this example demonstrates that we can get more accurate solutions by expanding Taylor series up to higher orders during the numerical computation.

Example 5.2. Consider Problem 5.1 with

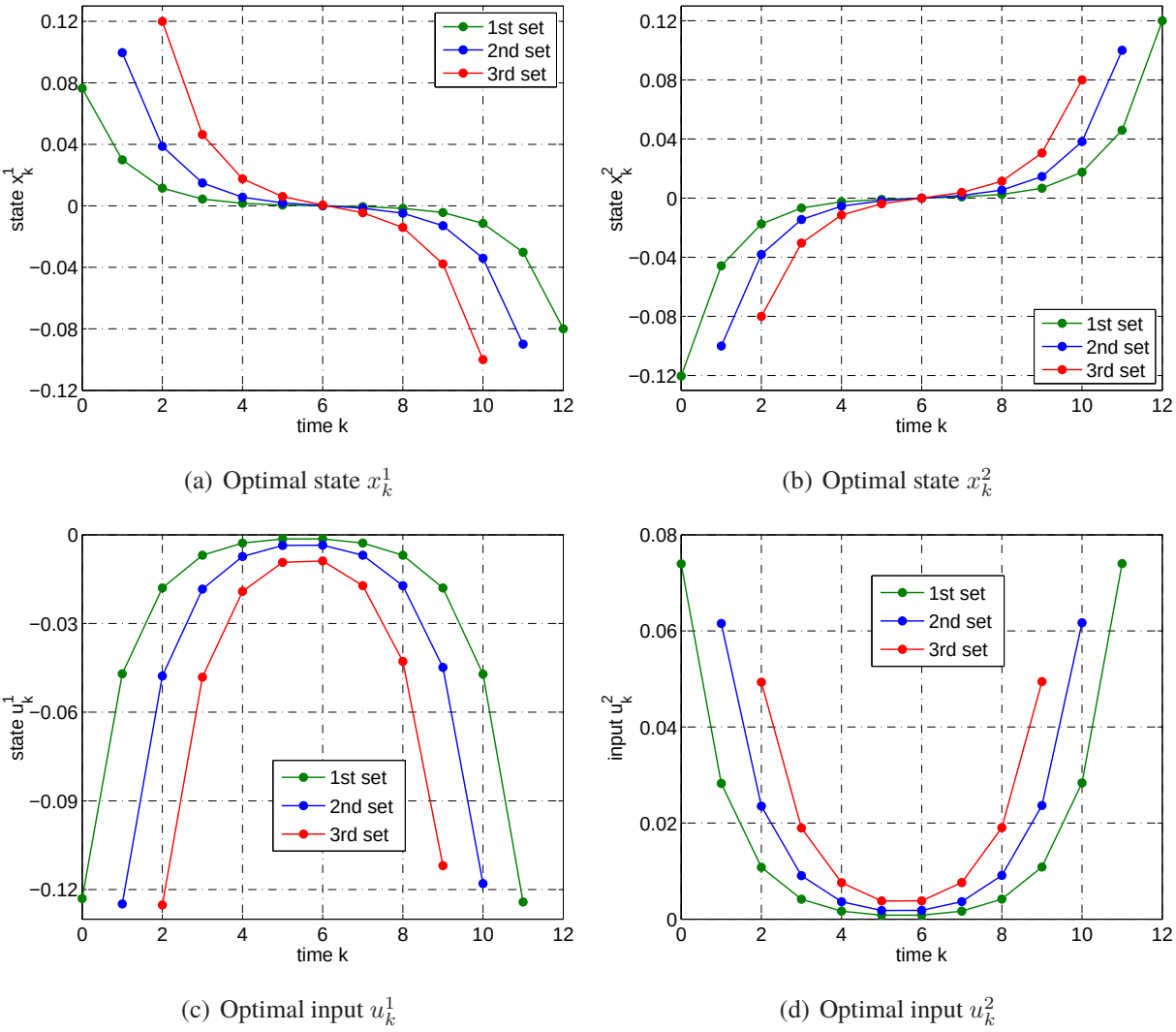
$$
\begin{aligned}
A(x_k) &= \begin{bmatrix} x_k^1 - x_k^2 + (x_k^1 - x_k^2)^2/20 + x_k^2 \exp(x_k^2/25) \\ x_k^2 \exp(x_k^2/25) \end{bmatrix}, & B(x_k) &= \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \\
Q(x_k) &= (x_k^1)^2/2 - x_k^1 x_k^2 + (x_k^2)^2, & R(x_k) &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}
\end{aligned}
$$

and three different sets of boundary conditions as in Table 5.4.

[†]If we choose the forward time calculation, it is opposite that the initial points will coincide with the given value, while state sequences separate along the forward time steps.

Table 5.4: Three different sets of initial-terminal jointed boundary conditions, where the first number separated by the comma in each cell is the initial/terminal time value, while the vector later is the initial/terminal boundary condition.

	Initial time and boundary condition	Terminal time and boundary condition
1st set	$0, [0.08, -0.12]^{\mathsf{T}}$	$12, [-0.08, 0.12]^{\mathsf{T}}$
2nd set	$1, [0.10, -0.10]^{\mathsf{T}}$	$11, [-0.09, 0.10]^{\mathsf{T}}$
3rd set	$2, [0.12, -0.08]^{\mathsf{T}}$	$10, [-0.10, 0.08]^{\mathsf{T}}$



(a) Optimal state x_k^1

(b) Optimal state x_k^2

(c) Optimal input u_k^1

(d) Optimal input u_k^2

Figure 5.1: Optimal state and input for three different sets of boundary conditions.

We apply Algorithms 5.1 and 5.2 to this problem. First in the off-line part, we expand Taylor series up to the 4th order. Based on this, we choose the time interval as the maximum, from 0 to 12, according to Table 5.4 to calculate the generating function coefficients. Then, optimal solutions corresponding to each particular set of boundary conditions can be efficiently generated

in the on-line part.

Results are presented in Figure 5.1, where Figure 5.1(a) and (b) are the first and second elements of the optimal state, and Figure 5.1(c) and (d) are the first and second elements of the optimal input, respectively. It can be found from Figure 5.1 that trajectories of the optimal state satisfy each set of boundary conditions in Table 5.4. The off-line and on-line computational time is 0.0068 [s] and 0.0024 [s] according to Algorithms 5.1 and 5.2, respectively. This example illustrates the effectiveness of the developed generating function method that it can solve such problems efficiently, especially when there is a large number of different sets of boundary conditions.

5.4 Summary

We develop the generating function method for the discrete-time nonlinear optimal control problems, including the presentation of the analytically optimal solutions and the exhibition of the Taylor series based numerical implementations. In the analytical part, we give the optimal input as the state feedforward control in terms of the generating functions. In the numerical part, due to the employed tensor notations, it is best for us to deeply investigate the Hamilton–Jacobi equation and prove some important properties, including the linearity and the recursiveness. This finally gives optimal solutions expressed only in terms of the pre-computed generating function coefficients and state boundary conditions by the Taylor series techniques. From this viewpoint, the developed generating function method is useful for on-demand optimal solutions generation for different boundary conditions.

Appendix

Proof of Theorem 5.2

Proof. (i) To prove this part, we first exhibit the explicit expression of the power series (5.13)

$$\begin{aligned}
&\left(\Phi\left(x_{k-1},\lambda_N;\mathscr{X}_{(\cdot,\cdot),k},\mathscr{F}_{(\cdot,\cdot),k}\right)\right)^{\ell_1}\\
=&x_k^{\ell_1}-\left(\mathscr{A}_{(0)}^{\ell_1}+\mathscr{A}_{(1)}^{\ell_1\ell_2}x_{k-1}^{\ell_2}+\mathscr{A}_{(2)}^{\ell_1\ell_2\ell_3}x_{k-1}^{\ell_2}x_{k-1}^{\ell_3}+\cdots\right)\\
&+\left(\mathscr{G}_{(0)}^{\ell_1\ell_2}+\mathscr{G}_{(1)}^{\ell_1\ell_2\ell_3}x_{k-1}^{\ell_3}+\mathscr{G}_{(2)}^{\ell_1\ell_2\ell_3\ell_4}x_{k-1}^{\ell_3}x_{k-1}^{\ell_4}+\cdots\right)\\
&\cdot\left(\mathscr{F}_{(1,0),k}^{\ell_2}+2\mathscr{F}_{(2,0),k}^{\ell_2\ell_3'}x_k^{\ell_3'}+\mathscr{F}_{(2,1),k}^{\ell_2\ell_3'}\lambda_N^{\ell_3'}+3\mathscr{F}_{(3,0),k}^{\ell_2\ell_3'\ell_4'}x_k^{\ell_3'}x_k^{\ell_4'}+\cdots\right)\\
=&0,\quad\forall\ell_1=1,2,\cdots,n
\end{aligned}\tag{5.25}$$

where $x_k^{\cdot}$ takes the form of (5.12). Then by collecting the constant terms from (5.25), we obtain the equation for $\mathscr{X}_{(0,0),k}^{\cdot}$

$$\begin{aligned}
&\left(\Phi\left(\mathscr{X}_{(\cdot,\cdot),k},\mathscr{F}_{(\cdot,\cdot),k}\right)\right)_{(0,0)}^{\ell_1}\\
=&\mathscr{X}_{(0,0),k}^{\ell_1}+\underbrace{\left(\mathscr{G}_{(0)}^{\ell_1\ell_2}\mathscr{F}_{(1,0),k}^{\ell_2}-\mathscr{A}_{(0)}^{\ell_1}\right)}_{=0\text{ by Lemmas 5.2(i) and 5.4}}+\mathscr{G}_{(0)}^{\ell_1\ell_2}\cdot\left(2\mathscr{F}_{(2,0),k}^{\ell_2\ell_3}+3\mathscr{F}_{(3,0),k}^{\ell_2\ell_3\ell_4}\mathscr{X}_{(0,0),k}^{\ell_4}+\cdots\right)\cdot\mathscr{X}_{(0,0),k}^{\ell_3}
\end{aligned}$$

$$=0, \quad \ell_1 = 1, 2, \cdots, n$$

which is (5.14) for the case $(i, j) = (0, 0)$. By solving the above n equations for the n variables $\mathscr{X}^{\ell_1}_{(0,0),k}$, $\forall \ell_1 = 1, 2, \cdots, n$, we obtain $\mathscr{X}^{\ell_1}_{(i,j),k} = 0$, $\forall \ell_1 = 1, 2, \cdots, n$ and $\forall k = 0, 1, \cdots, N$.

Note that this proof is based on Lemma 5.4 which is addressed later than this Theorem 5.2. But it can be found that Lemma 5.4 is independent on Theorem 5.2 from its proof. In other words, without Theorem 5.2, we can still prove Lemma 5.4. Hence we employ Lemma 5.4 here.

(ii)–(iii) To prove these two parts, we first show the linearity of the (i, j)-th equation in (5.14) with respect to the coefficient variable $\mathscr{X}^{\cdots}_{(i,j),k}$, $\forall (i, j) = (1, 0), (1, 1), \cdots$. For the sake of clarity, we here consider four cases with respect to the order (i, j), i.e. the cases $(i, j) = (1, 0)$, $(1, 1)$, (i, i), and the other orders. We first investigate the fourth case. By collecting terms with the same variable $\underbrace{\dot{x}_{k-1} \cdots \dot{x}_{k-1}}_{i-j} \underbrace{\dot{\lambda}_N \cdots \dot{\lambda}_N}_{j}$ from (5.25) (where $\mathscr{X}^{\cdots}_{(0,0),k} = 0$ based on (i)), we have

$$\begin{aligned}
&\left(\Phi\left(\mathscr{X}_{(\cdot,\cdot),k}, \mathscr{F}_{(\cdot,\cdot),k}\right)\right)^{\ell_1\ell_2\cdots\ell_{i+1}}_{(i,j)} \underbrace{x^{\ell_2}_{k-1} \cdots x^{\ell_{i-j+1}}_{k-1}}_{i-j} \underbrace{\lambda^{\ell_{i-j+2}}_N \cdots \lambda^{\ell_{i+1}}_N}_{j} \\
=&\mathscr{X}^{\ell_1\ell_2\cdots\ell_{i+1}}_{(i,j),k} \underbrace{x^{\ell_2}_{k-1} \cdots x^{\ell_{i-j+1}}_{k-1}}_{i-j} \underbrace{\lambda^{\ell_{i-j+2}}_N \cdots \lambda^{\ell_{i+1}}_N}_{j} \\
&+ 2\mathscr{G}^{\ell_1\ell'_2}_{(0)} \mathscr{F}^{\ell'_2\ell'_3}_{(2,0),k} \mathscr{X}^{\ell'_3\ell_2\cdots\ell_{i+1}}_{(i,j),k} \underbrace{x^{\ell_2}_{k-1} \cdots x^{\ell_{i-j+1}}_{k-1}}_{i-j} \underbrace{\lambda^{\ell_{i-j+2}}_N \cdots \lambda^{\ell_{i+1}}_N}_{j} \\
&+ \Psi_2\left(x_{k-1}, \lambda_N; \mathscr{X}_{(1,0),k}, \cdots, \mathscr{X}_{(i-1,j),k}, \mathscr{F}_{(2,0),k}, \cdots, \mathscr{F}_{(i+1,j),k}\right), \\
&\ell_1 = 1, 2, \cdots, n
\end{aligned} \tag{5.26}$$

where Ψ_2 denotes the analytic expression. Note that the corresponding coefficients $\mathscr{X}^{\cdots}_{(i,j),k}$ only exist in the first two terms on the right hand side of (5.26), while the third term Ψ_2 only contains coefficients with orders less than (i, j). Now, it is easy to know that the coefficient in (5.26) vanishes

$$\begin{aligned}
&\left(\Phi\left(\mathscr{X}_{(\cdot,\cdot),k}, \mathscr{F}_{(\cdot,\cdot),k}\right)\right)^{\ell_1\ell_2\cdots\ell_{i+1}}_{(i,j)} \\
=&\mathscr{X}^{\ell_1\ell_2\cdots\ell_{i+1}}_{(i,j),k} + 2\mathscr{G}^{\ell_1\ell'_2}_{(0)} \mathscr{F}^{\ell'_2\ell'_3}_{(2,0),k} \mathscr{X}^{\ell'_3\ell_2\cdots\ell_{i+1}}_{(i,j),k} \\
&+ \left(\Psi_2\left(\mathscr{X}_{(1,0),k}, \cdots, \mathscr{X}_{(i-1,j),k}; \mathscr{F}_{(2,0),k}, \cdots, \mathscr{F}_{(i+1,j),k}\right)\right)^{\ell_1\ell_2\cdots\ell_{i+1}} \\
=&0
\end{aligned}$$

where Ψ_2 denotes the analytic expression. The above equation is (5.14) for the fourth case. The equation (5.14) for the first three cases $(i, j) = (1, 0)$, $(1, 1)$, and (i, i) can also be obtained in similar ways. Now we exhibit the explicit expression of (5.14) for all these four cases together in the following (the corresponding coefficient variables are underlined for clarity)

$$\begin{cases} \underline{\mathscr{X}^{\ell_1\ell_2}_{(1,0),k}}+2\mathscr{G}^{\ell_1\ell_2'}_{(0)}\mathscr{F}^{\ell_2'\ell_3'}_{(2,0),k}\underline{\mathscr{X}^{\ell_3'\ell_2}_{(1,0),k}}+\Big(\mathscr{G}^{\ell_1\ell_2'\ell_2}_{(1)}\mathscr{F}^{\ell_2'}_{(1,0),k}-\mathscr{A}^{\ell_1\ell_2}_{(1)}\Big)=0, & (i,j)=(1,0) \quad (5.27)\\ \underline{\mathscr{X}^{\ell_1\ell_2}_{(1,1),k}}+2\mathscr{G}^{\ell_1\ell_2'}_{(0)}\mathscr{F}^{\ell_2'\ell_3'}_{(2,0),k}\underline{\mathscr{X}^{\ell_3'\ell_2}_{(1,1),k}}+\mathscr{G}^{\ell_1\ell_2'}_{(0)}\mathscr{F}^{\ell_2'\ell_2}_{(2,1),k}=0, & (i,j)=(1,1) \quad (5.28)\\ \underline{\mathscr{X}^{\ell_1\ell_2\cdots\ell_{i+1}}_{(i,i),k}}+2\mathscr{G}^{\ell_1\ell_2'}_{(0)}\mathscr{F}^{\ell_2'\ell_3'}_{(2,0),k}\underline{\mathscr{X}^{\ell_3'\ell_2\cdots\ell_{i+1}}_{(i,i),k}} & \\ \quad+\big(\Psi_1\big(\mathscr{X}_{(1,1),k},\cdots,\mathscr{X}_{(i-1,i-1),k};\mathscr{F}_{(2,0),k},\cdots,\mathscr{F}_{(i+1,i),k}\big)\big)^{\ell_1\ell_2\cdots\ell_{i+1}}=0, & (i,j)=(i,i) \quad (5.29)\\ \underline{\mathscr{X}^{\ell_1\ell_2\cdots\ell_{i+1}}_{(i,j),k}}+2\mathscr{G}^{\ell_1\ell_2'}_{(0)}\mathscr{F}^{\ell_2'\ell_3'}_{(2,0),k}\underline{\mathscr{X}^{\ell_3'\ell_2\cdots\ell_{i+1}}_{(i,j),k}} & \\ \quad+\big(\Psi_2\big(\mathscr{X}_{(1,0),k},\cdots,\mathscr{X}_{(i-1,j),k};\mathscr{F}_{(2,0),k},\cdots,\mathscr{F}_{(i+1,j),k}\big)\big)^{\ell_1\ell_2\cdots\ell_{i+1}}=0, & \text{other orders} \quad (5.30)\end{cases}$$

$\forall \ell_1,\ell_2,\cdots,\ell_{i+1}=1,2,\cdots,n$, where Ψ_1 denotes the analytic expression. According to the structure presented in the above, it is clear that each (i,j)-th equation is linear in the corresponding coefficient variable $\mathscr{X}^{\cdots}_{(i,j),k}$, $\forall (i,j)=(0,0),(1,0),\cdots$.

Second, notice (5.30), which has n^{i+1} equations for n^{i+1} coefficient variables $\mathscr{X}^{\ell_1\ell_2\cdots\ell_{i+1}}_{(i,j),k}$, $\forall \ell_1,\ell_2,\cdots,\ell_{i+1}=1,2,\cdots,n$. In addition, except the corresponding coefficient variable $\mathscr{X}_{(i,j),k}$, equations in (5.30) also contain the coefficient variables $\mathscr{X}_{(\cdot,\cdot),k}$ with orders less than (i,j). Due to this, we can determine all the variables $\mathscr{X}_{(\cdot,\cdot),k}$ recursively from the least order coefficient $\mathscr{X}_{(1,0),k}$, which can be expressed in terms of $\mathscr{F}_{(1,0),k}$ and $\mathscr{F}_{(2,0),k}$ according to (5.27). Based on these two points, we can uniquely determine these coefficient variables as analytic expressions in terms of $\mathscr{F}_{(\cdot,\cdot),k}$

$$\begin{aligned}&\mathscr{X}^{\ell_1\ell_2\cdots\ell_{i+1}}_{(i,j),k}=\Big(\Omega_1\big(\mathscr{F}_{(1,0),k},\mathscr{F}_{(2,0),k},\cdots,\mathscr{F}_{(i+1,j),k}\big)\Big)^{\ell_1\ell_2\cdots\ell_{i+1}},\\ &\forall \ell_1,\ell_2,\cdots,\ell_{i+1}=1,2,\cdots,n\end{aligned}$$

for the case $(i,j)=$ other orders, i.e. $(i,j)=(i,0),\cdots,(i,i-1)$ where $i\neq 0$, which is (5.15) for Theorem 5.2(ii). Here, Ω_1 denotes the analytic expression. The case $(i,j)=(i,i)$, $i\neq 0$, is also the same. According to (5.28) and (5.29), we conclude that the coefficient variables can also be uniquely determined as analytic expressions in terms of $\mathscr{F}_{(\cdot,\cdot),k}$

$$\begin{aligned}&\mathscr{X}^{\ell_1\ell_2\cdots\ell_{i+1}}_{(i,j),k}=\Big(\Omega_2\big(\mathscr{F}_{(2,0),k},\mathscr{F}_{(2,1),k},\cdots,\mathscr{F}_{(i+1,j),k}\big)\Big)^{\ell_1\ell_2\cdots\ell_{i+1}},\\ &\forall \ell_1,\ell_2,\cdots,\ell_{i+1}=1,2,\cdots,n\end{aligned}$$

which is (5.16) for Theorem 5.2(iii). Here, Ω_2 denotes the analytic expression. □

Proof of Theorem 5.3

Proof. To prove this theorem, we will show the procedure from (5.18) to (5.19) in detail. First, recall the expanded Hamilton–Jacobi equation (5.17), we re-exhibit it here and its six terms are marked with Terms I, II, ⋯, VI, respectively

$$\Gamma\left(x_{k-1},\lambda_N;\mathscr{F}_{(\cdot,\cdot),k-1},\mathscr{F}_{(\cdot,\cdot),k}\right)$$

$$
\begin{aligned}
&= \left(\mathscr{F}_{(0,0),k-1} + \mathscr{F}^{\ell_1}_{(1,0),k-1} x^{\ell_1}_{k-1} + \mathscr{F}^{\ell_1}_{(1,1),k-1} \lambda^{\ell_1}_N + \mathscr{F}^{\ell_1\ell_2}_{(2,0),k-1} x^{\ell_1}_{k-1} x^{\ell_2}_{k-1} + \cdots\right) \\
&\quad - \left(\mathscr{Q}_{(0)} + \mathscr{Q}^{\ell_1}_{(1)} x^{\ell_1}_{k-1} + \mathscr{Q}^{\ell_1\ell_2}_{(2)} x^{\ell_1}_{k-1} x^{\ell_2}_{k-1} + \mathscr{Q}^{\ell_1\ell_2\ell_3}_{(3)} x^{\ell_1}_{k-1} x^{\ell_2}_{k-1} x^{\ell_3}_{k-1} + \cdots\right) \\
&\quad - \left(\mathscr{F}^{\ell_1}_{(1,0),k} + 2\mathscr{F}^{\ell_1\ell_2}_{(2,0),k} x^{\ell_2}_k + \mathscr{F}^{\ell_1\ell_2}_{(2,1),k} \lambda^{\ell_2}_N + 3\mathscr{F}^{\ell_1\ell_2\ell_3}_{(3,0),k} x^{\ell_2}_k x^{\ell_3}_k + \cdots\right) \\
&\quad \cdot \left(\mathscr{A}^{\ell_1}_{(0)} + \mathscr{A}^{\ell_1\ell'_2}_{(1)} x^{\ell'_2}_{k-1} + \mathscr{A}^{\ell_1\ell'_2\ell'_3}_{(2)} x^{\ell'_2}_{k-1} x^{\ell'_3}_{k-1} + \mathscr{A}^{\ell_1\ell'_2\ell'_3\ell'_4}_{(3)} x^{\ell'_2}_{k-1} x^{\ell'_3}_{k-1} x^{\ell'_4}_{k-1} + \cdots\right) \\
&\quad + \frac{1}{2}\left(\mathscr{G}^{\ell_1\ell_2}_{(0)} + \mathscr{G}^{\ell_1\ell_2\ell_3}_{(1)} x^{\ell_3}_{k-1} + \mathscr{G}^{\ell_1\ell_2\ell_3\ell_4}_{(2)} x^{\ell_3}_{k-1} x^{\ell_4}_{k-1} + \cdots\right) \\
&\quad \cdot \left(\mathscr{F}^{\ell_1}_{(1,0),k} + 2\mathscr{F}^{\ell_1\ell'_2}_{(2,0),k} x^{\ell'_2}_k + \mathscr{F}^{\ell_1\ell'_2}_{(2,1),k} \lambda^{\ell'_2}_N + +3\mathscr{F}^{\ell_1\ell'_2\ell'_3}_{(3,0),k} x^{\ell'_2}_k x^{\ell'_3}_k + \cdots\right) \\
&\quad \cdot \left(\mathscr{F}^{\ell_2}_{(1,0),k} + 2\mathscr{F}^{\ell_2\ell''_3}_{(2,0),k} x^{\ell''_3}_k + \mathscr{F}^{\ell_2\ell''_3}_{(2,1),k} \lambda^{\ell''_3}_N + 3\mathscr{F}^{\ell_2\ell''_3\ell''_4}_{(3,0),k} x^{\ell''_3}_k x^{\ell''_4}_k + \cdots\right) \\
&\quad - \left(\mathscr{F}_{(0,0),k} + \mathscr{F}^{\ell_1}_{(1,0),k} x^{\ell_1}_k + \mathscr{F}^{\ell_1}_{(1,1),k} \lambda^{\ell_1}_N + \mathscr{F}^{\ell_1\ell_2}_{(2,0),k} x^{\ell_1}_k x^{\ell_2}_k + \cdots\right) \\
&\quad + \left(\mathscr{F}^{\ell_1}_{(1,0),k} + 2\mathscr{F}^{\ell_1\ell_2}_{(2,0),k} x^{\ell_2}_k + \mathscr{F}^{\ell_1\ell_2}_{(2,1),k} \lambda^{\ell_2}_N + 3\mathscr{F}^{\ell_1\ell_2\ell_3}_{(3,0),k} x^{\ell_2}_k x^{\ell_3}_k + \cdots\right) \cdot x^{\ell_1}_k \\
&= 0
\end{aligned}
\tag{5.31}
$$

where $\dot{x_k}$ takes the form of (5.12) with coefficients $\mathscr{X}_{(\cdot,\cdot),k}$ expressed in terms of $\mathscr{F}_{(\cdot,\cdot),k}$ according to Theorem 5.2 (where $\mathscr{X}_{(0,0),k} = 0$). We collect terms with the same variable from (5.31), and list them term by term sequentially from I to VI in the following, where zero coefficients are marked with underline, and $\mathscr{F}_{(\cdot,\cdot),k}$ with orders greater than (i,j) with double underlines (note that some $\mathscr{X}_{(\cdot,\cdot),k}$ also contain great order $\mathscr{F}_{(\cdot,\cdot),k}$ according to Theorem 5.2, hence they are also double underlined)

- From Term I: $\mathscr{F}^{\ell_1\ell_2\cdots\ell_i}_{(i,j),k-1} \underbrace{x^{\ell_1}_{k-1}\cdots x^{\ell_{i-j}}_{k-1}}_{i-j} \underbrace{\lambda^{\ell_{i-j+1}}_N \cdots \lambda^{\ell_i}_N}_{j}$
- From Term II: 0
- From Term III:

$$
\begin{aligned}
&\underline{\mathscr{A}^{\ell_1}_{(0)}} \cdot \Bigg(2\mathscr{F}^{\ell_1\ell_2}_{(2,0),k} \underline{\underline{\mathscr{X}^{\ell_2\ell_3\cdots\ell_{i+2}}_{(i,j),k}}} \underbrace{x^{\ell_3}_{k-1}\cdots x^{\ell_{i-j+2}}_{k-1}}_{i-j} \underbrace{\lambda^{\ell_{i-j+3}}_N \cdots \lambda^{\ell_{i+2}}_N}_{j} + \cdots \\
&\qquad + (i+1-j)\cdot \underline{\underline{\mathscr{F}^{\ell_1\ell_2\cdots\ell_{i+1}}_{(i+1,j),k}}} \underbrace{\mathscr{X}^{\ell_2\ell'_3}_{(1,0),k} x^{\ell'_3}_{k-1} \cdots \mathscr{X}^{\ell_{i-j+1}\ell'_{i-j+2}}_{(1,0),k} x^{\ell'_{i-j+2}}_{k-1}}_{i-j} \underbrace{\lambda^{\ell_{i-j+2}}_N \cdots \lambda^{\ell_{i+1}}_N}_{j}\Bigg) \\
&+ \mathscr{A}^{\ell_1\ell_2}_{(1)} x^{\ell_2}_{k-1} \cdot \Bigg(2\mathscr{F}^{\ell_1\ell'_2}_{(2,0),k} \mathscr{X}^{\ell'_2\ell'_3\cdots\ell'_{i+1}}_{(i-1,j),k} \underbrace{x^{\ell'_3}_{k-1}\cdots x^{\ell'_{i-j+1}}_{k-1}}_{i-j-1} \underbrace{\lambda^{\ell'_{i-j+2}}_N \cdots \lambda^{\ell'_{i+1}}_N}_{j} + \cdots \\
&\qquad + (i-j)\cdot \mathscr{F}^{\ell_1\ell'_2\cdots\ell'_i}_{(i,j),k} \underbrace{\mathscr{X}^{\ell'_2\ell''_3}_{(1,0),k} x^{\ell''_3}_{k-1} \cdots \mathscr{X}^{\ell'_{i-j}\ell''_{i-j+1}}_{(1,0),k} x^{\ell''_{i-j+1}}_{k-1}}_{i-j-1} \underbrace{\lambda^{\ell'_{i-j+1}}_N \cdots \lambda^{\ell'_i}_N}_{j}\Bigg) \\
&+ \cdots
\end{aligned}
$$

- From Term IV:

$$
\frac{1}{2}\mathscr{G}^{\ell_1\ell_2}_{(0)} \cdot \Bigg(\underline{\mathscr{F}^{\ell_1}_{(1,0),k}} \cdot \Bigg(2\mathscr{F}^{\ell_2\ell'_3}_{(2,0),k} \underline{\underline{\mathscr{X}^{\ell'_3\ell'_4\cdots\ell'_{i+3}}_{(i,j),k}}} \underbrace{x^{\ell'_4}_{k-1}\cdots x^{\ell'_{i-j+3}}_{k-1}}_{i-j} \underbrace{\lambda^{\ell'_{i-j+4}}_N \cdots \lambda^{\ell'_{i+3}}_N}_{j} + \cdots
$$

$$+(i+1-j)\cdot\underline{\underline{\mathscr{F}^{\ell_2\ell'_3\cdots\ell'_{i+2}}_{(i+1,j),k}}}\underbrace{\mathscr{X}^{\ell'_3\ell''_4}_{(1,0),k}x^{\ell''_4}_{k-1}\cdots\mathscr{X}^{\ell'_{i-j+2}\ell''_{i-j+3}}_{(1,0),k}x^{\ell''_{i-j+3}}_{k-1}}_{i-j}\underbrace{\lambda^{\ell'_{i-j+3}}_N\cdots\lambda^{\ell'_{i+2}}_N}_{j}\Bigg)$$

$$+2\mathscr{F}^{\ell_1\ell'_2}_{(2,0),k}\cdot\Bigg(\underline{\underline{\mathscr{X}^{\ell'_2\ell''_3\cdots\ell''_{i+2}}_{(i,j),k}}}\underbrace{x^{\ell''_3}_{k-1}\cdots x^{\ell''_{i-j+2}}_{k-1}}_{i-j}\underbrace{\lambda^{\ell''_{i-j+3}}_N\cdots\lambda^{\ell''_{i+2}}_N}_{j}\underline{\mathscr{F}^{\ell_2}_{(1,0),k}}+\cdots$$

$$+(i-j)\cdot\mathscr{X}^{\ell'_2\ell''_3}_{(1,0),k}x^{\ell''_3}_{k-1}\mathscr{F}^{\ell_2\ell'_3\cdots\ell'_{i+1}}_{(i,j),k}\underbrace{\mathscr{X}^{\ell'_3\ell''_4}_{(1,0),k}x^{\ell''_4}_{k-1}\cdots\mathscr{X}^{\ell'_{i-j+1}\ell''_{i-j+2}}_{(1,0),k}x^{\ell''_{i-j+2}}_{k-1}}_{i-j}\underbrace{\lambda^{\ell'_{i-j+2}}_N\cdots\lambda^{\ell'_{i+1}}_N}_{j}\Bigg)$$

$$+\cdots\Bigg)$$

$$+\frac{1}{2}\mathscr{G}^{\ell_1\ell_2\ell_3}_{(1)}x^{\ell_3}_{k-1}\cdot\Bigg(\mathscr{F}^{\ell_1}_{(1,0),k}\cdot\Bigg(2\mathscr{F}^{\ell_2\ell'_3}_{(2,0),k}\mathscr{X}^{\ell'_3\ell'_4\cdots\ell'_{i+2}}_{(i-1,j),k}\underbrace{x^{\ell'_4}_{k-1}\cdots x^{\ell'_{i-j+2}}_{k-1}}_{i-j-1}\underbrace{\lambda^{\ell'_{i-j+3}}_N\cdots\lambda^{\ell'_{i+2}}_N}_{j}+\cdots$$

$$+(i-j)\cdot\mathscr{F}^{\ell_2\ell'_3\cdots\ell'_{i+1}}_{(i,j),k}\underbrace{\mathscr{X}^{\ell'_3\ell''_4}_{(1,0),k}x^{\ell''_4}_{k-1}\cdots\mathscr{X}^{\ell'_{i-j+1}\ell''_{i-j+2}}_{(1,0),k}x^{\ell''_{i-j+2}}_{k-1}}_{i-j}\underbrace{\lambda^{\ell'_{i-j+2}}_N\cdots\lambda^{\ell'_{i+1}}_N}_{j}\Bigg)$$

$$+\cdots\Bigg)$$

$$+\cdots$$

- From Term V:
$$\underline{\mathscr{F}^{\ell_1}_{(1,0),k}}\,\underline{\underline{\mathscr{X}^{\ell_1\ell'_2\cdots\ell'_{i+1}}_{(i,j),k}}}\underbrace{x^{\ell'_2}_{k-1}\cdots x^{\ell'_{i-j+1}}_{k-1}}_{i-j}\underbrace{\lambda^{\ell'_{i-j+2}}_N\cdots\lambda^{\ell'_{i+1}}_N}_{j}+\cdots$$
$$+\mathscr{F}^{\ell_1\ell_2\cdots\ell_i}_{(i,j),k}\underbrace{\mathscr{X}^{\ell_1\ell'_2}_{(1,0),k}x^{\ell'_2}_{k-1}\cdots\mathscr{X}^{\ell_{i-j}\ell'_{i-j+1}}_{(1,0),k}x^{\ell'_{i-j+1}}_{k-1}}_{i-j}\underbrace{\lambda^{\ell_{i-j+1}}_N\cdots\lambda^{\ell_i}_N}_{j}$$

- From Term VI:
$$\underline{\mathscr{F}^{\ell_1}_{(1,0),k}}\,\underline{\underline{\mathscr{X}^{\ell_1\ell'_2\cdots\ell'_{i+1}}_{(i,j),k}}}\underbrace{x^{\ell'_2}_{k-1}\cdots x^{\ell'_{i-j+1}}_{k-1}}_{i-j}\underbrace{\lambda^{\ell'_{i-j+2}}_N\cdots\lambda^{\ell'_{i+1}}_N}_{j}+\cdots$$
$$+(i-j)\cdot\mathscr{F}^{\ell_1\ell'_2\cdots\ell'_i}_{(i,j),k}\underbrace{\mathscr{X}^{\ell'_2\ell''_3}_{(1,0),k}x^{\ell''_3}_{k-1}\cdots\mathscr{X}^{\ell'_{i-j}\ell''_{i-j+1}}_{(1,0),k}x^{\ell''_{i-j+1}}_{k-1}}_{i-j-1}\underbrace{\lambda^{\ell'_{i-j+1}}_N\cdots\lambda^{\ell_i\prime}_N}_{j}\mathscr{X}^{\ell_1\ell'''_2}_{(1,0),k}x^{\ell'''_{k-1}}_{k-1}.$$

By eliminating the variable $\underbrace{x^{\cdot}_{k-1}\cdots x^{\cdot}_{k-1}}_{i-j}\underbrace{\lambda^{\cdot}_N\cdots\lambda^{\cdot}_N}_{j}$ from the above six expressions, and letting their summation equal to zero, we obtain the difference equations (5.19). This is the whole procedure we derive difference equations for generating function coefficients.

Second, notice the coefficients $\mathscr{F}_{(i+1,0),k},\cdots,\mathscr{F}_{(i+1,j),k}$ in the above six expressions. Since their orders are greater than (i,j), hence they are double underlined. As mentioned before, note that $\mathscr{X}_{(i,j),k}$ in the above expressions also contains $\mathscr{F}_{(i+1,j),k}$ according to Theorem 5.2, hence $\mathscr{X}_{(i,j),k}$ is also double underlined. Importantly, it should be noted that these double underlined coefficients only exist in the presented rows above. In other words, there is no such coefficient existing in the unpresented rows. Further, it can be found from the above expressions that each great order coefficient ($\mathscr{F}_{(i+1,\cdot),k}$ or $\mathscr{X}_{(i,j),k}$) is multiplied by the underlined $\mathscr{F}_{(1,0),k}=0$ (Lemma 5.4) or $\mathscr{A}_{(0)}=0$ (5.2(i)). They are the two unique ways that these great order coefficients exist in the above expressions. This implies the difference equations in (5.19) can be solved recursively for the generating function coefficients $\mathscr{F}_{(i,j),k}$ with respect to the Taylor series order index (i,j).

The above is for the case $(i,j)=(i,1),(i,2),\cdots,(i,i-1)$. We can also prove the recur-

siveness for the case $(i, j) = (i, 0)$ and (i, i). Since the proofs can be stated in a similar way to the first case, hence we omit them here. □

Chapter 6

Conclusion

This thesis has three main contributions in extending the generating function method to solve continuous-time state constrained problems, developing the double generating functions method for discrete-time LQ optimal control with numerical stability analysis of the optimal generators, and solving the discrete-time nonlinear optimal control problem via generating functions.

Chapter 2 introduces preliminaries of the Hamiltonian system and the generating functions. For both the continuous and discrete time optimal control problems, we give necessary and sufficient (only for continuous-time case) conditions for optimality, derive Hamilton–Jacobi equations and generating functions, provide optimal solutions (only for continuous-time case), present relations between the generating function and the value function, and exhibit the LQ cases.

Chapter 3 extends the generating function method to the LQ optimal control problems with path and terminal state constraints by employing penalties. The penalized problem with a general penalty is introduced to approximate the original constrained problem. We show that both of them are convex problems and optimal solutions of the penalized problem will converge to the ones of the original constrained problem when the penalty factor approaches zero. Moreover, a recursive condition is presented to eliminate the coupling relation between the generating function coefficients with lower and higher indices in the ordinary differential equations so that they can be solved recursively. Based on this, we also summarize how to design penalties that is suitable for the generating function method, and give an algorithm presents how to generate optimal solutions repetitively for different boundary conditions. This framework is able to give accurate solutions, and also possesses the significance in online repetitive computation for different boundary conditions.

Chapter 4 presents a whole framework of double generating functions method to the discrete-time LQ optimal control problem, including the development of generators for optimal solutions and the numerical stability analysis. We first derive the discrete forward and backward single generating functions by solving appropriate right and left Hamilton–Jacobi equations based on necessary conditions for optimality, and give six generators for optimal solutions based on double generating functions constructed by selecting any two different single generating functions among the candidates. Then, under the invertibility analysis of the inverse terms in these generators based on properties of the coefficients presented in this chapter, we conclude that the generators constructed by double generating functions with opposite time directions are available

for applications under some mild conditions, while the generators with the same time directions should be avoided for real practice. This numerical stability analysis can also be generalized to the existing single generating function method.

Chapter 5 develops the generating function method for the discrete-time nonlinear optimal control problems, including the presentation of the analytical solutions and the exhibition of the Taylor series based numerical implementations. In the analytical part, we give the optimal input as the state feedforward control in terms of the generating functions. In the numerical part, due to the employed tensor notations, it is best for us to deeply investigate the Hamilton–Jacobi equation and prove some important properties, including the linearity and the recursiveness. This finally gives optimal solutions expressed only in terms of the pre-computed generating function coefficients and state boundary conditions by the Taylor series techniques. From this viewpoint, the developed generating function method is useful for the on-demand optimal solutions generation for different boundary conditions.

The generating function method exhibits theoretical insights in solving optimal control problems and practical implication for real world applications. After one decade of the development, there still has significant potentiality in its further research. Future work includes the study of convergence region of the Taylor series to Hamilton–Jacobi equation and the extension of the generating function method to solve stochastic optimal control problems.

Bibliography

[1] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, Cambridge, 2004.

[2] A.C. Chiang, *Elements of Dynamic Optimization*, Waveland Press, Illinois, 1992.

[3] A.E. Bryson, *Dynamic Optimization*, Pearson Education, New Jersey, 1998.

[4] J. Engwerda, *LQ Dynamic Optimization and Differential Games*, John Wiley & Sons, New Jersey, 2005.

[5] M.I. Kamien and N.L. Schwartz, *Dynamic Optimization, Second Edition: The Calculus of Variations and Optimal Control in Economics and Management*, Dover Publications, New York, 2012.

[6] L.S. Pontryagin, V.G. Boltyanskii, R.V. Gamkrelidze and E.F. Mishchenko, *The Mathematical Theory of Optimal Processes*, Interscience Publishers, New York, 1962.

[7] R.E. Bellman, *Dynamic Programming*, Princeton University Press, New Jersey, 1957.

[8] H. Kwakernaak and R. Sivan, *Linear Optimal Control Systems*, Wiley Interscience, New Jersey, 1972.

[9] L.D. Berkovitz, *Optimal Control Theory*, Springer-Verlag, New York, 1974.

[10] A.E. Bryson and Y.C. Ho, *Applied Optimal Control: Optimization, Estimation and Control*, Taylor & Francis Group, London, 1975.

[11] W.H. Fleming and R.W. Rishel, *Deterministic and Stochastic Optimal Control*, Springer-Verlag, Berlin, 1982.

[12] R.F. Stengel, *Optimal Control and Estimation*, Dover Publications, New York, 1994.

[13] K. Zhou, J.C. Doyle and K. Glover, *Robust and Optimal Control*, Pearson Education, New Jersey, 1995.

[14] F.L. Lewis and V.L. Syrmos, *Optimal Control, Second Edition*, John Wiley & Sons, New Jersey, 1995.

[15] D.E. Kirk, *Optimal Control Theory: An Introduction*, Dover Publications, New York, 2004.

[16] M. Athans and P.L. Falb, *Optimal Control: An Introduction to the Theory and Its Applications*, Dover Publications, New York, 2006.

[17] T.L. Friesz, *Dynamic Optimization and Differential Games*, Springer, New York, 2010.

[18] D. Liberzon, *Calculus of Variations and Optimal Control Theory: A Concise Introduction*, Princeton University Press, New Jersey, 2012.

[19] D.P. Bertsekas, *Dynamic Programming and Optimal Control*, Athena Scientific, New Hampshire, 2012.

[20] B.D.O. Anderson and J.B. Moore, *Optimal Control: Linear Quadratic Methods*, Dover Publications, New York, 2014.

[21] L.S. Lasdon, A.D. Waren and R.K. Rice, An interior penalty method for inequality constrained optimal control problems, *IEEE Transactions on Automatic Control*, vol. 12, no. 4, pp. 388–395, 1967.

[22] L. Lasdon, S. Mitter and A. Waren, The conjugate gradient method for optimal control problems, *IEEE Transactions on Automatic Control*, vol. 12, no. 2, pp. 132–138, 1967.

[23] J. Willems, Least squares stationary optimal control and the algebraic Riccati equation, *IEEE Transactions on Automatic Control*, vol. 16, no. 6, pp. 621–634, 1971.

[24] C.J. Goh and K.L. Teo, Control parametrization: A unified approach to optimal control problems with general constraints, *Automatica*, vol. 24, no. 1, pp. 3–18, 1988.

[25] O. von Stryk, Numerical solution of optimal control problems by direct collocation, *International Series of Numerical Mathematics*, vol. 111, pp. 129–143, 1993.

[26] G. Elnagar, M.A. Kazemi and M. Razzaghi, The pseudospectral Legendre method for discretizing optimal control problems, *IEEE Transactions on Automatic Control*, vol. 40, no. 10, pp. 1793–1796, 1995.

[27] A. Bemporad and M. Morari, Control of systems integrating logic, dynamics, and constraints, *Automatica*, vol. 35, no. 3, pp. 407–427, 1999.

[28] A. Rantzer and M. Johansson, Piecewise linear quadratic optimal control, *IEEE Transactions on Automatic Control*, vol. 45, no. 4, pp. 629–637, 2000.

[29] J.A.K. Suykens, J. Vandewalle and B. De Moor, Optimal control by least squares support vector machines, *Neural Networks*, vol. 14, no. 1, pp. 23–35, 2001.

[30] F. Fahroo and I.M. Ross, Costate Estimation by a Legendre Pseudospectral Method, *Journal of Guidance, Control, and Dynamics*, vol. 24, no. 2, pp. 270–277, 2001.

[31] A. Bemporad, M. Morari, V. Dua and E.N. Pistikopoulos, The explicit linear quadratic regulator for constrained systems, *Automatica*, vol. 38, no. 1, pp. 3–20, 2002.

[32] F. Fahroo and I.M. Ross, Direct Trajectory Optimization by a Chebyshev Pseudospectral Method, *Journal of Guidance, Control, and Dynamics*, vol. 25, no. 1, pp. 160–166, 2002.

[33] D. Carlson, A. Haurie and A. Leizarowitz, *Infinite Horizon Optimal Control*, Springer-Verlag, Berlin, 1991.

[34] E.F. Camacho and C. Bordons, *Model Predictive Control*, Springer-Verlag, New York, 1999.

[35] V.M. Guibout and D.J. Scheeres, Solving relative two-point boundary value problems: Spacecraft formulation flight transfers application, *Journal of Guidance, Control, and Dynamics*, vol. 27, no. 4, pp. 693–704, 2004.

[36] C. Park and D. J. Scheeres, Determination of optimal feedback terminal controllers for general boundary conditions using generating functions, *Automatica*, vol. 42, no. 5, pp. 869–875, 2006.

[37] C. Park, D.J. Scheeres, V.M. Guibout and A. Bloch, Global solution for the optimal feedback control of the underactuated Heisenberg system, *IEEE Transactions on Automatic Control*, vol. 53, no. 11, pp. 2638–2642, 2008.

[38] Z. Hao, K. Fujimoto and Y. Hayakawa, Optimal trajectory generation for linear systems based on double generating functions, in *Proceedings of the 51st IEEE Conference on Decision and Control*, pp. 3827–3832, 2012.

[39] Z. Hao, K. Fujimoto and Y. Hayakawa, Optimal Trajectory Generation for Linear Systems Based on Double Generating Functions, *SICE Journal of Control, Measurement, and System Integration*, vol. 6, no. 3, pp. 194–201, 2013.

[40] Z. Hao, K. Fujimoto, and Y. Hayakawa, Optimal trajectory generation for nonlinear systems based on double generating functions, in *Proceedings of 2013 American Control Conference*, pp. 6382–6387, 2013.

[41] M. Bando and H. Yamakawa, A new optimal orbit control for two-point boundary-value problem using generating functions, *Advances in the Astronautical Sciences*, vol. 134, pp. 245–260, 2009.

[42] Z. Hao and K. Fujimoto, Approximate solutions to the Hamilton–Jacobi equations for generating functions with a quadratic cost function with respect to the input, in *Proceedings of the 4th IFAC Workshop on Lagrangian and Hamiltonian Methods for Nonlinear Control*, pp. 194–199, 2012.

[43] Z. Hao, K. Fujimoto, and Y. Hayakawa, Approximate solutions to the Hamilton–Jacobi equations for generating functions: The general cost function case, in *Proceedings of the 9th Asian Control Conference*, pp. 1–6, 2013.

[44] Z. Wu and W. Zhong, A structure-preserving algorithm for the minimum H_∞ norm computation of finite-time state feedback control problem, *International Journal of Control*, vol. 82, no. 4, pp. 773–781, 2009.

[45] H. Peng, Q. Gao, Z. Wu and W. Zhong, Efficient sparse approach for solving receding-horizon control problems, *Journal of Guidance, Control, and Dynamics*, vol. 36, no. 6, pp. 1864–1872, 2013.

[46] Y. Okura and K. Fujimoto, A new framework of robust LQ optimal control for parameter variation and its application to the double generating functions method, in *Proceedings of the 54th IEEE Conference on Decision and Control*, pp. 3236–3241, 2015.

[47] Z. Wu and M. Mesbahi, Symplectic transformation based analytical and numerical methods for linear quadratic control with hard terminal constraints, *SIAM Journal on Control and Optimization*, vol. 50, no. 2, pp. 652–671, 2012.

[48] T. Ohsawa, A. M. Bloch and M. Leok, Discrete Hamilton–Jacobi theory, *SIAM Journal on Control and Optimization*, vol. 49, no. 4, pp. 1829–1856, 2011.

[49] T. Lee, Discrete-time optimal feedback control via Hamilton-Jacobi theory with an application to hybrid systems, in *Proceedings of the 51st IEEE Conference on Decision and Control*, pp. 7055–7062, 2012.

[50] T. Lee, Optimal control of partitioned hybrid systems via discrete-time Hamilton–Jacobi theory, *Automatica*, vol. 50, no. 8, pp. 2062–2069, 2014.

[51] V.M. Guibout and D.J. Scheeres, Spacecraft formation dynamics and design, *Journal of Guidance, Control, and Dynamics*, vol. 29, no. 1, pp. 121–133, 2006.

[52] C. Park, V.M Guibout, and D.J. Scheeres, "Solving optimal continuous thrust rendezvous problems with generating functions," *Journal of Guidance, Control, and Dynamics*, vol. 29, no. 2, pp. 321–331, 2006.

[53] M. Bando and H. Yamakawa, Low-thrust trajectory optimization using second-order generating functions, in *Proceedings of the SICE Annual Conference 2010*, pp. 804–810, 2010.

[54] Z. Hao, K. Fujimoto and Y. Hayakawa, Application of the double generating function method to optimal gait generation for a biped robot, in *Proceedings of the 32nd Chinese Control Conference*, pp. 2338–2343, 2013.

[55] Z. Hao, K. Fujimoto and Y. Hayakawa, On-demand optimal gait generation for a compass biped robot based on the double generating function method, in *Proceedings of the 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3108–3113, 2013.

[56] Z. Hao, K. Fujimoto and Y. Hayakawa, Optimal gait generation for a compass biped robot via the double generating functions method, *SICE Journal of Control, Measurement, and System Integration*, vol. 7, no. 2, pp. 96–103, 2014.

[57] L.S. Lasdon, A.D. Waren and R.K. Rice, An interior penalty method for inequality constrained optimal control problems, *IEEE Transactions on Automatic Control*, vol. 12, no. 4, pp. 388–395, 1967.

[58] P. Malisani, F. Chaplais and N. Petit, An interior penalty method for optimal control problems with state and input constraints of nonlinear systems, *Optimal Control Applications and Methods*, vol. 37, no. 1, pp. 3–33, 2016.

[59] J.F. Bonnans and T. Guilbaud, Using Logarithmic penalties in the shooting algorithm for optimal control problems, *Optimal Control Applications and Methods*, vol. 24, no. 5, pp. 257–278, 2003.

[60] V. Kucera, The discrete Riccati equation of optimal control, *Kybernetika*, vol. 8, no. 5, pp. 430–447, 1972.

[61] H.B. Keller, *Numerical Methods for Two-Point Boundary-Value Problems*, Dover Publications, New Jersey, 1968.

[62] B. Chachuat, *Nonlinear and Dynamic Optimization: From Theory to Practice*, Automatic Control Laboratory, EPFL, Switzerland, 2007.

[63] H. Goldstein, C.P. Poole Jr. and J.L. Safko, *Classical Mechanics*, Addison-Wesley, Boston, 2001.

[64] A.V. Fiacco and G.P. McCormick, *Nonlinear Programming: Sequential Unconstrained Minimization* , John Wiley & Sons, New Jersey, 1968.

[65] J. Huang and W.J. Rugh, Stabilization on zero-error manifolds and the nonlinear servomechanism problem, *IEEE Transactions on Automatic Control*, vol. 37, no. 7, pp. 1009–1013, 1992.

[66] Z. Hao, *Optimal Trajectory Generation via Double Generating Functions and Application to Biped Robots*, Ph.D. Thesis, Nagoya University, Nagoya, Japan, 2014.

[67] W.H. Clohessy, Terminal guidance system for satellite rendezvous, *Journal of the Aerospace Sciences*, vol. 27, no. 9, pp. 653–658, 1960.

[68] R.R. Bitmead, M.R. Gevers, I.R. Petersen and R.J. Kaye, Monotonicity and stabilizability properties of solutions of the Riccati difference equation: Propositions, lemmas, theorems, fallacious conjectures and counterexamples, *Systems & Control Letters*, vol. 5, no. 5, pp. 309–315, 1985.

[69] P.E. Caines and D.Q. Mayne, On the discrete time matrix Riccati equation of optimal control, *International Journal of Control*, vol. 12, no. 5, pp. 785–794, 1970.

[70] J.E. Marsden and T.S. Ratiu, *Introduction to Mechanics and Symmetry*, Springer-Verlag, New York, 1999.

Published papers

Chapter 3

- D. Chen, K. Fujimoto and T. Suzuki, Generating function approach to linear quadratic optimal control problem with constraints on the state, in *Proceedings of the 53rd IEEE Conference on Decision and Control*, pp. 6659–6664, 2014.
- D. Chen, K. Fujimoto and T. Suzuki, Optimal gait generation of constrained compass biped robot via generating function approach, in *Proceedings of the SICE Annual Conference 2015*, pp. 626–631, 2015.

Chapter 4

- D. Chen, Z. Hao, K. Fujimoto and T. Suzuki, Discrete-time linear quadratic optimal control via double generating functions, *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E98-A, no. 3, pp. 833–842, 2015.
- D. Chen, Z. Hao, K. Fujimoto and T. Suzuki, Discrete-time linear quadratic optimal control via forward generating functions, in *Proceedings of the SICE Annual Conference 2013*, pp. 1475–1478, 2013.
- D. Chen, Z. Hao, K. Fujimoto and T. Suzuki, Discrete-time linear quadratic optimal control with fixed and free terminal state via double generating functions, in *Proceedings of the 19th IFAC World Congress*, pp. 6044–6049, 2014.

Chapter 5

- D. Chen, K. Fujimoto and T. Suzuki, Discrete-time nonlinear optimal control via generating functions, to appear in *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, 2016.
- D. Chen, K. Fujimoto and T. Suzuki, Solving discrete-time nonlinear optimal control problem by generating function approach, in *Proceedings of the SICE Annual Conference 2014*, pp. 1055–1058, 2014.
- D. Chen, K. Fujimoto and T. Suzuki, Double generating function approach to discrete-time nonlinear optimal control problems, in *Proceedings of the 54th IEEE Conference on Decision and Control*, pp. 3894–3899, 2015.