

The Computational Kleinman-Newton Method in Solving Nonlinear Nonquadratic Control Problems

Jinghong Kang

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

Doctor of Philosophy
in
Mathematics

David L. Russell, Chair
Tao Lin
Jong Uhn Kim
Robert C. Rogers
Shu-ming Sun

Blacksburg, Virginia

April 1998

Keywords: Nonlinear Nonquadratic Control, Hamiltonian Function, Adjoint Equation, Fixed Point Theorem, Contraction, Interpolation

The Computational Kleinman-Newton Method in Solving Nonlinear Nonquadratic Control Problems

By

Jinghong Kang

Committee Chair: David L. Russell

Department of Mathematics

(Abstract)

This thesis deals with non-linear non-quadratic optimal control problems in an autonomous system and a related iterative numerical method, the Kleinman-Newton method, for solving the problem. The thesis proves the local convergence of Kleinman-Newton method using the contraction mapping theorem and then describes how this Kleinman-Newton method may be used to numerically solve for the optimal control and the corresponding solution. In order to show the proof and the related numerical work, it is necessary to review some of earlier work in the beginning of Chapter 1 [Zhang], and to introduce the Kleinman-Newton method at the end of the chapter. In Chapter 2 we will demonstrate the proof. In Chapter 3 we will show the related numerical work and results.

Acknowledgements

I would like to express my sincere gratitude to my advisor, Dr. David Russell, for his valuable advice, support, and direction, which made the completion of this dissertation possible. I also want to thank Dr. Jong Kim, Dr. Tao Lin, Dr. Robert Rogers, and Dr. Shu-Ming Sun for their serving on my committee. They were always supportive and encouraging.

Contents

| | |
|---|----|
| CHAPTER 1. INTRODUCTION | 1 |
| 1.1 AN OPTIMAL CONTROL PROBLEM..... | 1 |
| 1.2 THE NECESSARY CONDITIONS FOR THE OPTIMAL CONTROL PROBLEM | 3 |
| 1.3 LINEAR QUADRATIC OPTIMAL CONTROL | 8 |
| 1.4 NONLINEAR-NONQUADRATIC CONTROL SYSTEM..... | 10 |
| 1.5 <i>Kleinman-Newton Method for Linear Quadratic Problem</i> | 12 |
| 1.6 <i>Extension of the Kleinman-Newton Method to the Nonlinear Nonquadratic Case</i> | 14 |
| CHAPTER 2. THE PROOF OF CONVERGENCE OF THE KLEINMAN-NEWTON METHOD | 17 |
| 2.1 THE ITERATION FORMULA OF THE KLEINMAN-NEWTON METHOD | 17 |
| 2.2 THE RELATED HAMILTONIAN FUNCTION AND ADJOINT EQUATION..... | 18 |
| 2.3 THE PROOF OF THE CONVERGENCE OF (2.2.17)..... | 22 |
| CHAPTER 3. USING THE KLEINMAN-NEWTON METHOD TO COMPUTE THE OPTIMAL CONTROL | 30 |
| 3.1 ASSUMPTIONS..... | 30 |
| 3.2 COMPUTATIONAL ALGORITHM FOR THE KLEINMAN-NEWTON METHOD | 33 |
| <i>Algorithm Step I. Partition the Domain D into Finite Number of Elements</i> | 33 |
| <i>Algorithm Step II. Initial Stabilizing Control</i> | 34 |
| <i>Algorithm Step III. Start the Iteration</i> | 35 |
| <i>Algorithm Step IV. Conclusion of the Iteration</i> | 36 |
| 3.3 COMPUTATIONAL EXAMPLES | 37 |
| 3.3.1 <i>Pendulum System</i> | 37 |
| 3.3.2 <i>Application of Kleinman-Newton Method</i> | 38 |
| 3.3.3 <i>Iteration of Kleinman-Newton Method</i> | 39 |
| 3.3.4 <i>Interpolation of Control Values</i> | 41 |
| 3.3.5 <i>A Brief Description of Data Structure</i> | 42 |
| 3.3.6 <i>Computation Result</i> | 44 |
| 3.4 CONCLUSION | 58 |
| REFERENCE..... | 59 |

List of Figures

FIGURE 1.2.1: THE OPTIMAL CONTROL $\hat{u}(t)$ IS PIECEWISE CONTINUOUS WITH RESPECT TO t , AND BOTH $\tau + \delta$ AND τ ARE POINTS OF CONTINUITY OF $\hat{u}(t)$ 4

FIGURE 1.2.2: THE TRAJECTORIES OF $\hat{x}(t)$ AND $\tilde{x}(t)$ FOR $t \in [t_0, t_1]$ 4

FIGURE 3.1.1: EACH VALUE OF $V(x(t_i))$ IS FOUND BY INTEGRATING BACKWARDS ALONG THE TRAJECTORY $x(t)$ WITH THE INITIAL VALUE $x(t_i)$ 33

FIGURE 3.3.1: ALL THE GRID POINTS IN THE CHOSEN DOMAIN D 38

FIGURE 3.3.2: THE PATTERN OF THE VALUES OF V AT ITERATION STEP K 40

FIGURE 3.3.3: THERE ARE 4 CONTROL VALUES SURROUNDING A SOLUTION VALUE $x(n)$ 41

FIGURE 3.3.4: TWO LAYERS OF THE WHOLE DATA STRUCTURE: THE NODE IS LINKED WITH ONE ANOTHER WITHIN THE SAME LAYER, AND EACH LAYER IS LINKED WITH ONE ANOTHER BY THE CORNER NODE..... 43

FIGURE 3.3.5: OUTPUT OF $V(x)$ IN ITERATION 0. 45

FIGURE 3.3.6: OUTPUT OF $V(x)$ IN ITERATION 1. 45

FIGURE 3.3.7: OUTPUT OF $V(x)$ IN ITERATION 2. 46

FIGURE 3.3.8: OUTPUT OF $V(x)$ IN ITERATION 3. 46

FIGURE 3.3.9: OUTPUT OF $V(x)$ IN ITERATION 4. 47

FIGURE 3.3.10: OUTPUT OF $V(x)$ IN ITERATION 5. 47

FIGURE 3.3.11: OUTPUT OF $V(x)$ IN ITERATION 6. 48

FIGURE 3.3.12: OUTPUT OF $V(x)$ IN ITERATION 7. 48

FIGURE 3.3.13: OUTPUT OF $\mu(x)$ IN ITERATION 0..... 49

FIGURE 3.3.14: OUTPUT OF $\mu(x)$ IN ITERATION 1..... 49

FIGURE 3.3.15: OUTPUT OF $\mu(x)$ IN ITERATION 2..... 50

| | |
|--|----|
| FIGURE 3.3.16: OUTPUT OF $\mu(x)$ IN ITERATION 3..... | 50 |
| FIGURE 3.3.17: OUTPUT OF $\mu(x)$ IN ITERATION 4..... | 51 |
| FIGURE 3.3.18: OUTPUT OF $\mu(x)$ IN ITERATION 5..... | 51 |
| FIGURE 3.3.19: OUTPUT OF $\mu(x)$ IN ITERATION 6..... | 52 |
| FIGURE 3.3.20: OUTPUT OF $\mu(x)$ IN ITERATION 7..... | 52 |
| FIGURE 3.3.21: THE PHASE PLANE WITH INITIAL VALUES OF $(-0.5, -1)$ AND $(0.5, 1.3)$.. | 54 |
| FIGURE 3.3.22: THE PHASE PLANE WITH INITIAL VALUES OF $(-0.5, 1.2)$ AND $(0.5, -1.1)$ | 54 |
| FIGURE 3.3.23: CONTROL $\mu(x(t))$ CORRESPONDS TO THE INITIAL VALUE OF $(-0.5, -1)$. | 55 |
| FIGURE 3.3.24: CONTROL $\mu(x(t))$ CORRESPONDS TO THE INITIAL VALUE OF $(0.5, 1.3)$... | 55 |
| FIGURE 3.3.25: CONTROL $\mu(x(t))$ CORRESPONDS TO THE INITIAL VALUE OF $(-0.5, 1.2)$. | 56 |
| FIGURE 3.3.26: CONTROL $\mu(x(t))$ CORRESPONDS TO THE INITIAL VALUE OF $(0.5, -1.1)$. | 56 |

LIST OF TABLES

| | |
|--|----|
| TABLE 3.3.1: THE COST OF THE FOUR TRAJECTORIES CALCULATED ACCORDING TO THE DEFINITION (3.3.4)..... | 57 |
| TABLE 3.3.2: THE COMPARISON BETWEEN THE FOUR RESULTED THEORETICAL COST AND NUMERICAL COST. | 57 |

Chapter 1. Introduction

1.1 An Optimal Control Problem

A control system can be represented by the differential equation

$$\dot{x} = F(t, x, u), \quad (1.1.1)$$

where $F(t, x, u)$ is continuously differentiable with respect to $x \in R^n$ and $u \in R^m$, and piecewise continuous with respect to t . Though in some cases, the domain of F may be further restricted, here we will not consider that case. The control system is called autonomous system, if t in (1.1.1) is implicit, i.e., the function $F(t, x, u)$ takes the form of $F(x, u)$.

Usually control systems are categorized into linear and nonlinear systems. A linear control system can be obtained by linearizing $F(t, x, u)$ into the form of (1.1.2), that is taking partial derivatives with respect to x and u respectively, and only keeping the linear terms.

$$\dot{x} = A(t)x + B(t)u, \quad (1.1.2)$$

where $A(t)$ and $B(t)$ are $n \times n$ and $n \times m$ matrices, and they are continuous with respect to t .

In order to define an optimal control problem, we first assume that system performance can be described in terms of non-negative cost functions $G(t, x, u)$ and

$G_1(x)$. We suppose the cost function, $G(t, x, u)$, is continuously differentiable with respect to x and u , and piecewise continuous with respect to t , and we let $G_1(x)$ be continuously differentiable with respect to x . The optimal control problem is to minimize the cost functional

$$\int_{t_0}^{t_1} G(t, x(t), u(t)) dt + G_1(x(t_1)), \quad (1.1.3)$$

for fixed $t_1 > t_0$, assuming an initial state $x(0) = x_0 \in R^n$, by appropriate choice of the control function $u(t)$ on $[t_0, t_1]$. The related concepts on observability, stability, and controllability for both linear and nonlinear control systems may be found in [Russell] and [Lee & Markus].

As for the autonomous linear quadratic problem, the differential equation will be as follows

$$\dot{x} = F(x, u) = Ax + Bu, \quad (1.1.4)$$

where $A = \frac{\partial F(0,0)}{\partial x}$ and $B = \frac{\partial F(0,0)}{\partial u}$ are real matrices. The corresponding cost function is

$$G(x, u) = \begin{pmatrix} x^* & u^* \end{pmatrix} \begin{pmatrix} W & R \\ R^* & U \end{pmatrix} \begin{pmatrix} x \\ u \end{pmatrix}. \quad (1.1.5)$$

where W , U , and R are real matrices, and $\begin{pmatrix} W & R \\ R^* & U \end{pmatrix}$ is a real symmetric positive definite matrix. Since t is implicit in (1.1.4) and (1.1.5), $F(x, u)$ and $G(x, u)$ are linear and quadratic terms of x and u respectively, the control problem is called an autonomous linear quadratic problem. The details on finding the linear optimal control u (analytically and numerically) are in [Russell].

The problem being studied here is an extension of the autonomous linear quadratic problem, i.e., autonomous nonlinear nonquadratic problem (briefly described by the acronym NLNQ). The differential equation is

$$\dot{x} = F(x, u) = Ax + Bu + f(x, u), \quad (1.1.6)$$

where $f(x, u)$ consists of higher order (≥ 2) remainder terms in x and u , with $f(0,0)=0$, $\frac{\partial f(0,0)}{\partial x}=0$, and $\frac{\partial f(0,0)}{\partial u}=0$. The corresponding cost function is

$$G(x, u) = \begin{pmatrix} x^* & u^* \end{pmatrix} \begin{pmatrix} W & R \\ R^* & U \end{pmatrix} \begin{pmatrix} x \\ u \end{pmatrix} + g(x, u), \quad (1.1.7)$$

where $g(x, u)$ consists of higher order (≥ 3) remainder terms in x and u , with $g(0,0)=0$, $\frac{\partial g(0,0)}{\partial x}=0$, $\frac{\partial g(0,0)}{\partial u}=0$, $\frac{\partial^2 g(0,0)}{\partial x^2}=0$, and $\frac{\partial^2 g(0,0)}{\partial u^2}=0$. The matrices A and B are assumed to be a stabilizable pair, that is, there exists a real matrix K for which $A + BK$ is a stability matrix whose eigenvalues all have negative real parts.

1.2 The Necessary Conditions for the Optimal Control Problem

We begin by reviewing the necessary conditions (cf. [Lee & Markus], [Lukes]), corresponding to the ‘‘Maximum Principle’’ (the details about ‘‘Maximum Principle’’ are in [Pontryagin], [Lee & Markus]), which must be satisfied by the optimal control \hat{u} .

We use $\hat{u}(t)$ to denote an assumed optimal control and $\hat{x}(t)$ to denote the corresponding optimal trajectory. In order to develop necessary conditions characterizing the optimal control $\hat{u}(t)$ and the corresponding response $\hat{x}(t)$, we need to assume that the control $\hat{u}(t)$ is piecewise continuous with respect to t . Further, we need to construct a modified control $\tilde{u}(t)$ by selecting $\delta > 0$ and supposing $t_0 < \tau < \tau +$

$\delta < t_1$ where both $\tau + \delta$ and τ are points of continuity of $\hat{u}(t)$ (see Figure 1). Specifically, we let

$$\tilde{u}(t) = \begin{cases} \hat{u}(t) & \text{for } t \in [t_0, \tau) \cup (\tau + \delta, t_1] \\ \hat{u}(t) + \mu & \text{for } t \in [\tau, \tau + \delta] \end{cases} \quad (1.2.1)$$

where μ is an arbitrary constant vector in R^m . The scalar case is illustrated in Figure 1.

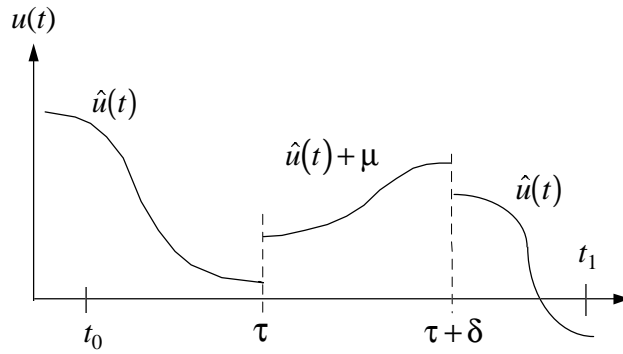


Figure 1.2.1: The optimal control $\hat{u}(t)$ is piecewise continuous with respect to t , and both $\tau + \delta$ and τ are points of continuity of $\hat{u}(t)$.

Let the corresponding response to $\tilde{u}(t)$ be $\tilde{x}(t)$; thus

$$\dot{\tilde{x}} = F(t, \tilde{x}(t), \tilde{u}(t)), \quad \tilde{x}(0) = x_0 \in R^n, \quad (1.2.2)$$

and set $\tilde{x} = \hat{x}(t) + \xi(t)$ with $\xi(\tau) = 0$ and $\xi(t) \equiv 0$, $0 \leq t \leq \tau$ (see Figure 2),

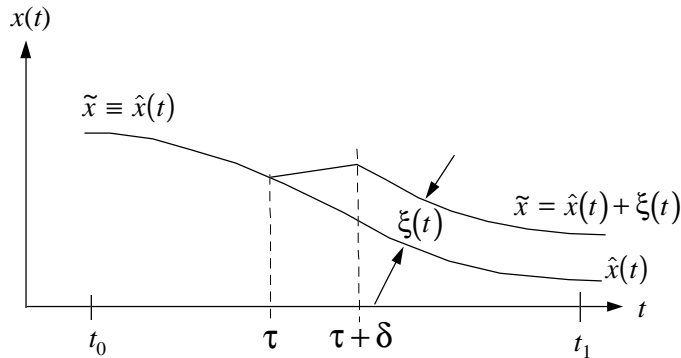


Figure 1.2.2: The trajectories of $\hat{x}(t)$ and $\tilde{x}(t)$ for $t \in [t_0, t_1]$.

we have

$$\tilde{x} = \hat{x}(t) + \xi(t), \quad (1.2.3)$$

and hence by (1.2.3), for t in $[\tau, \tau + \delta]$,

$$\begin{aligned} \dot{\xi}(t) &= \dot{\tilde{x}}(t) - \dot{\hat{x}}(t) = F(t, \hat{x} + \xi, \hat{u} + \mu) - F(t, \hat{x}, \hat{u}) \\ &= F(t, \hat{x} + \xi, \hat{u} + \mu) - F(t, \hat{x}, \hat{u} + \mu) + F(t, \hat{x}, \hat{u} + \mu) - F(t, \hat{x}, \hat{u}) \\ &= \left[\frac{\partial F(t, \hat{x}, \hat{u} + \mu)}{\partial x} \xi(t) + O(\delta) \right] + F(t, \hat{x}, \hat{u} + \mu) - F(t, \hat{x}, \hat{u}) \\ &= [A(t, \hat{u})\xi(t) + F(t, \hat{x}, \hat{u} + \mu) - F(t, \hat{x}, \hat{u})] + O(\delta), \end{aligned} \quad (1.2.4)$$

where $O(\delta)$ represents the higher order terms for x , $A(t, \hat{u}) = \frac{\partial F(t, \hat{x}, \hat{u} + \mu)}{\partial x}$, and

$\Phi(t, s, \hat{u})$ is the fundamental solution matrix of the differential equation

$$\frac{\partial \Phi(t, s, \hat{u})}{\partial t} = A(t, \hat{u})\Phi(t, s, \hat{u}), \quad (1.2.5)$$

such that the fundamental solution $\Phi(s, s, \hat{u}) = I$.

Thus, since $\xi(\tau) = 0$, the solution for (1.2.4) is given by

$$\begin{aligned} \xi(t) &= \int_{\tau}^t \Phi(t, s, \hat{u}) [F(s, \hat{x}(s), \hat{u}(s) + \mu) - F(s, \hat{x}(s), \hat{u}(s)) + O(\delta)] ds \\ &= (t - \tau) [F(\tau, \hat{x}(\tau), \hat{u}(\tau) + \mu) - F(\tau, \hat{x}(\tau), \hat{u}(\tau))] + O(\delta) \end{aligned} \quad (1.2.6)$$

in $\tau \leq t \leq \tau + \delta$, where $O(\delta)$ represents higher order terms for x , and thus

$$\xi(\tau + \delta) = \delta [F(\tau, \hat{x}(\tau), \hat{u}(\tau) + \mu) - F(\tau, \hat{x}(\tau), \hat{u}(\tau))] + O(\delta). \quad (1.2.7)$$

For $\tau + \delta \leq t \leq t_1$ we have

$$\begin{aligned} \dot{\xi}(t) &= F(t, \hat{x}(t) + \xi(t), \hat{u}) - F(t, \hat{x}(t), \hat{u}(t)) \\ &= A(t, \hat{u})\xi(t) + O(h) \end{aligned} \quad (1.2.8)$$

for $t \in [\tau + \delta, t_1]$, with the initial data at $\tau + \delta$ given by (1.2.7).

By (1.2.5) and (1.2.7), the solution for (1.2.8) on $t \in [\tau + \delta, t_1]$ is

$$\begin{aligned}
\xi(t) &= \Phi(t, \tau + \delta)\xi(\tau + \delta) + O(\delta) \\
&= \Phi(t, \tau + \delta)\left\{\delta\left[F(\tau, \hat{x}(\tau), \hat{u}(\tau) + \mu) - F(\tau, \hat{x}(\tau), \hat{u}(\tau))\right] + O(\delta)\right\} + O(\delta) \\
&= \left[\Phi(t, \tau) + O(\delta)\right]\left\{\delta\left[F(\tau, \hat{x}(\tau), \hat{u}(\tau) + \mu) - F(\tau, \hat{x}(\tau), \hat{u}(\tau))\right] + O(\delta)\right\} + O(\delta) \\
&= \delta\Phi(t, \tau)\left[F(\tau, \hat{x}(\tau), \hat{u}(\tau) + \mu) - F(\tau, \hat{x}(\tau), \hat{u}(\tau))\right] + O(\delta). \tag{1.2.9}
\end{aligned}$$

If we set $t_0 = \tau$, then from $\hat{u}(t)$ to $\tilde{u}(t) = \hat{u}(t) + \mu$ and from $\hat{x}(t)$ to $\tilde{x}(t) = \hat{x}(t) + \xi(t)$, the change in the cost is

$$\begin{aligned}
\Delta &= \left[\int_{\tau}^{t_1} G(t, \tilde{x}(t), \tilde{u}(t))dt + G_1(\tilde{x}(t_1))\right] - \left[\int_{\tau}^{t_1} G(t, \hat{x}(t), \hat{u}(t))dt + G_1(\hat{x}(t_1))\right] \\
&= \int_{\tau}^{t_1} \left[G(t, \tilde{x}(t), \tilde{u}(t)) - G(t, \hat{x}(t), \hat{u}(t))\right]dt + G_1(\tilde{x}(t_1)) - G_1(\hat{x}(t_1)) \\
&= \int_{\tau}^{t_1} \left[G(t, \hat{x}(t) + \xi(t), \hat{u}(t) + \mu) - G(t, \hat{x}(t), \hat{u}(t))\right]dt + G_1(\hat{x}(t_1) + \xi(t_1)) - G_1(\hat{x}(t_1)). \tag{1.2.10}
\end{aligned}$$

Since in (1.2.10) the integral term is

$$\begin{aligned}
&\int_{\tau}^{t_1} \left[G(t, \hat{x}(t) + \xi(t), \hat{u}(t) + \mu) - G(t, \hat{x}(t), \hat{u}(t))\right]dt \\
&= \int_{\tau}^{\tau + \delta} \left[G(t, \hat{x}(t) + \xi(t), \hat{u}(t) + \mu) - G(t, \hat{x}(t), \hat{u}(t))\right]dt + \int_{\tau + \delta}^{t_1} \left[G(t, \hat{x}(t) + \xi(t), \hat{u}(t) + \mu) - G(t, \hat{x}(t), \hat{u}(t))\right]dt \\
&= \delta \left[G(t, \hat{x}(t) + \xi(t), \hat{u}(t) + \mu) - G(t, \hat{x}(t), \hat{u}(t))\right] \\
&+ \delta \int_{\tau}^{t_1} \frac{\partial G(t, \hat{x}(t), \hat{u}(t))}{\partial x} \Phi(t, \tau, u) \left[F(\tau, \hat{x}(\tau), \hat{u}(\tau) + \mu) - F(\tau, \hat{x}(\tau), \hat{u}(\tau))\right]dt + O(\delta) \tag{1.2.11}
\end{aligned}$$

and the non-integral, final state, term is

$$G_1(\hat{x}(t_1) + \xi(t_1)) - G_1(\hat{x}(t_1)) = \delta \frac{\partial G_1(\hat{x}(t_1))}{\partial x} \Phi(t_1, \tau) \left[F(\tau, \hat{x}(\tau), \hat{u}(\tau) + \mu) - F(\tau, \hat{x}(\tau), \hat{u}(\tau))\right] + O(h), \tag{1.2.12}$$

then by (1.2.10), (1.2.11), and (1.2.12), the change of cost to the first order accuracy in δ is

$$\begin{aligned} \Delta = \delta & \left\{ \left[G(t, \hat{x}(t) + \xi(t), \hat{u}(t) + \mu) - G(t, \hat{x}(t), \hat{u}(t)) \right] \right. \\ & \left. + \left[\int_{\tau}^{t_1} \frac{\partial G(t, \hat{x}(t), \hat{u}(t))}{\partial x} \Phi(t, \tau, u) dt + \frac{\partial G_1(\hat{x}(t_1))}{\partial x} \Phi(t_1, u) \right] \left[F(\tau, \hat{x}(\tau), \hat{u}(\tau) + \mu) - F(t, \hat{x}(t), \hat{u}(t)) \right] \right\}. \end{aligned} \quad (1.2.13)$$

If we define the adjoint variable as

$$\Lambda(\tau)^* = \int_{\tau}^{t_1} \frac{\partial G(t, \hat{x}(t), \hat{u}(t))}{\partial x} \Phi(t, \tau, u) dt + \frac{\partial G_1(\hat{x}(t_1))}{\partial x} \Phi(t_1, \tau), \quad (1.2.14)$$

then (1.2.13) can be changed into

$$\begin{aligned} \Delta = \delta & \left\{ \left[G(t, \hat{x}(t) + \xi(t), \hat{u}(t) + \mu) - G(t, \hat{x}(t), \hat{u}(t)) \right] + \Lambda(\tau)^* \left[F(\tau, \hat{x}(\tau), \hat{u}(\tau) + \mu) - F(t, \hat{x}(t), \hat{u}(t)) \right] \right\} \\ & = \delta \left\{ \left[G(t, \hat{x}(t) + \xi(t), \hat{u}(t) + \mu) + \Lambda(\tau)^* F(\tau, \hat{x}(\tau), \hat{u}(\tau) + \mu) \right] - \left[\Lambda(\tau)^* F(t, \hat{x}(t), \hat{u}(t)) + G(t, \hat{x}(t), \hat{u}(t)) \right] \right\}. \end{aligned} \quad (1.2.15)$$

If we further define the Hamiltonian function as

$$H(t, x(t), u(t), \Lambda(t)) = \Lambda(t)^* F(t, x(t), u(t)) + G(t, x(t), u(t)), \quad (1.2.16)$$

then (1.2.15) takes the form

$$\Delta = H(t, \hat{x}(t), \hat{u}(t) + \mu, \Lambda(t)) - H(\tau, \hat{x}(\tau), \hat{u}(\tau), \Lambda(\tau)). \quad (1.2.17)$$

Because of the assumption that the optimal control $\hat{u}(t)$ that minimizes the cost, the changes of the cost should be always non-negative, i.e. $\Delta \geq 0$, that is for any control $u(t)$ we always have

$$H(t, \hat{x}(t), u(t), \Lambda(t)) \geq H(t, \hat{x}(t), \hat{u}(t), \Lambda(t)). \quad (1.2.18)$$

Hence the necessary condition for optimal control problem (1.1.2) can be stated as

$$\min_u H(t, \hat{x}(t), u(t), \Lambda(t)) \quad (1.2.19)$$

where the adjoint function $\Lambda(t)$ in (1.2.14) can be re-interpreted as the solution satisfying the adjoint differential equation

$$\frac{d\Lambda(t)^*}{dt} = -\Lambda(t)^* \frac{\partial F(\hat{x}(t), \hat{u}(t))}{\partial x} - \frac{\partial G(\hat{x}(t), \hat{u}(t))}{\partial x} \quad (1.2.20)$$

with the terminal boundary condition $\Lambda(t_1)^* = \frac{\partial G_1(\hat{x}(t_1))}{\partial x}$.

In Section 1.4 we will see that the necessary condition (1.2.19) and (1.2.20) are also sufficient conditions for an optimal control and further, (1.2.18) actually can be restated as

$$\min_u H(t, \hat{x}(t), u(t), \Lambda(t)) = 0. \quad (1.2.21)$$

1.3 Linear Quadratic Optimal Control

If the control system has the autonomous form

$$\dot{x} = F(x(t), u(t)), \quad x(0) = x_0 \in R^n, \quad (1.3.1)$$

and if $x = 0, u = 0$ is an equilibrium point, i.e., $F(0,0) = 0$, then with our differentiability assumption, we can linearize $F(x(t), u(t))$ by use of the Taylor expansion to the first order accuracy with respect to x . We have the form

$$\dot{x} = Ax + Bu, \quad x(0) = x_0 \in R^n \quad (1.3.2)$$

which is called a *linear* control system. We take the classical regulator problem as an example which involves the quadratic cost function (cf. [Russell], [Lee & Markus])

$$G(x(t), u(t)) = \frac{1}{2} \begin{pmatrix} x^* & u^* \end{pmatrix} \begin{pmatrix} W & R \\ R^* & U \end{pmatrix} \begin{pmatrix} x \\ u \end{pmatrix}, \quad (1.3.3)$$

where the matrix $\begin{pmatrix} W & R \\ R^* & U \end{pmatrix}$ is assumed to be a symmetric positive definite matrix.

The related optimal control problem is then

$$\min \left\{ \frac{1}{2} \int_{t_0}^{t_1} \begin{pmatrix} x^* & u^* \end{pmatrix} \begin{pmatrix} W & R \\ R^* & U \end{pmatrix} \begin{pmatrix} x \\ u \end{pmatrix} dt + x(t_1)^* W_1 x(t_1) \right\}, \quad (1.3.4)$$

for which by (1.2.21), (1.3.2) and (1.3.3), the necessary condition is

$$\min \left\{ \Lambda(t)^* [Ax(t) + Bu(t)] + x(t)^* Wx(t) + u^* Rx + x^* Ru + u^* Uu \right\}. \quad (1.3.5)$$

Here the minimum is obtained by setting $\frac{\partial H(x(t), u(t), \Lambda(t))}{\partial u} = 0$; from which we obtain the optimal control $\hat{u}(t)$

$$B^* \Lambda(t) + R^* x(t) + U \hat{u}(t) = 0, \quad (1.3.6)$$

which yields

$$\hat{u}(t) = -U^{-1} [B^* \Lambda(t) + R^* x(t)]. \quad (1.3.7)$$

Substituting (1.3.7) into (1.3.2) we have

$$\dot{\hat{x}} = (A - BU^{-1}R^*)\hat{x}(t) - BU^{-1}B^*\Lambda(t), \quad \hat{x}(0) = x_0 \in R^n \quad (1.3.8)$$

By (1.2.20), we can have the adjoint equation

$$\frac{d\Lambda(t)^*}{dt} = -\Lambda(t)A - x(t)^*W - u^*R^* \quad (1.3.9)$$

with the boundary condition $\Lambda(t_1) = W_1 x(t_1)$.

In order to solve the equation (1.3.8) and (1.3.9), we let $\Lambda(t) = Q(t)x(t)$ and substitute it into (1.3.8) and (1.3.9) to obtain the so-called Riccati equation

$$\dot{Q}(t) + A^*Q(t) + Q(t)A + W - [Q(t)B + R]U^{-1}[B^*Q(t) + R^*] = 0 \quad (1.3.10)$$

with boundary value $Q(t_1) = W_1$. By solving (1.3.10) and (1.3.8), we obtain the related optimal control

$$\hat{u}(t) = \hat{K}(t)\hat{x}(t), \quad \text{where } \hat{K}(t) = -U^{-1}[B^*Q(t) + R^*]. \quad (1.3.11)$$

There are four important results for the constant coefficient linear quadratic control problem [Russell] and [Anderson & Moore]:

1) For constant coefficients A, B, W, R , and U , and cost defined on $[0, \infty)$, we can take

$Q(t) \equiv Q$ the unique symmetric positive definite solution of the matrix quadratic equation

$$A^*Q + QA + W - [QB + R]U^{-1}[B^*Q + R^*] = 0. \quad (1.3.12)$$

2) The control $\hat{u}(t) = -U^{-1}[B^*Q + R^*]\hat{x}(t) = \hat{K}\hat{x}(t)$ is the unique optimal control for the linear quadratic system, $\hat{K} = -U^{-1}[B^*Q + R^*]$ being the constant feedback matrix and Q the unique symmetric positive definite solution of (1.3.12).

3) $x_0^*Qx_0 = \int_0^\infty G(x, \hat{u})dt$ is the optimal cost.

4) $\hat{\Lambda}(t) = Q\hat{x}(t)$.

1.4 Nonlinear-Nonquadratic Control System

The problem of extending the linear quadratic control theory to nonlinear systems and nonquadratic costs, which may be abbreviated as the NLNQ problem, was studied by [Lukes]. The nonlinear-nonquadratic optimal control problem applies (1.1.1) when the system takes the form

$$\dot{x} = F(x(t), u(t)) = Ax(t) + Bu(t) + f(x(t), u(t)) \quad (1.4.1)$$

with initial conditions $x(0) = x_0 \in R^n$, $f(0,0) = 0$, $\frac{\partial f(0,0)}{\partial x} = 0$, and $\frac{\partial f(0,0)}{\partial u} = 0$, and

where A and B are constant. The corresponding cost functional is defined by letting $t_1 \rightarrow \infty$, i.e.

$$\begin{aligned} V(x(t), u(t)) &= \int_0^\infty G(x(t), u(t))dt \\ &= \int_0^\infty \left\{ \begin{pmatrix} x^* & u^* \end{pmatrix} \begin{pmatrix} W & R \\ R^* & U \end{pmatrix} \begin{pmatrix} x \\ u \end{pmatrix} + g(x(t), u(t)) \right\} dt, \end{aligned} \quad (1.4.2)$$

where $Ax + Bu$ and $\begin{pmatrix} x^* & u^* \end{pmatrix} \begin{pmatrix} W & R \\ R^* & U \end{pmatrix} \begin{pmatrix} x \\ u \end{pmatrix}$ are linear and quadratic leading terms of $F(x(t), u(t))$ and $G(x(t), u(t))$ respectively, $f(x, u)$ and $g(x, u)$ are their higher order terms satisfying $f(0,0)=0$, $\frac{\partial f(0,0)}{\partial x}=0$, and $\frac{\partial f(0,0)}{\partial u}=0$. As in the linear constant coefficient case, the cost function $G_1(x(t_1))$ is replaced by 0 here.

The admitted feedback controls for the nonlinear-nonquadratic system may be decomposed into linear and nonlinear parts, but we will not separate these two parts until Chapter 2 when we prove the convergence. The control can be written as

$$u(t) = K(x(t), \Lambda(t)). \quad (1.4.3)$$

In [Lukes] there is a theorem and a lemma which describe the existence and uniqueness results for the optimal control and a related functional equation in the nonlinear-nonquadratic control problem. The theorem and lemma are

Theorem 1. For the stabilizable control process in R^n ,

$$\dot{x} = Ax + Bu + f(x, u)$$

with the initial condition $x(0) = x_0$ near the origin in R^n and with the cost functional

$$V(x(t), u(t)) = \int_0^\infty G(x(t), u(t)) dt$$

there exists a unique optimal stabilizing feedback control \hat{u} which uniquely solves the functional equation

$$\frac{\partial V(x)}{\partial x} \frac{\partial F(x, u)}{\partial u} + \frac{\partial G(x, u)}{\partial u} = 0 \quad (1.4.4)$$

near the origin.

*Lemma 1. For each stabilizing control $u(x)$ there exists a positive invariant neighborhood N_u of the origin in R^n space wherein the cost function $V(x) = x^*Qx + v(x)$ with the initial condition x_0 and where $v(x)$ is the higher order term and Q is given by (1.3.11). In N_u the functional equation*

$$\frac{\partial V(x)}{\partial x} F(x, u) + G(x, u) = 0 \quad (1.4.5)$$

obtains.

Equations (1.4.4) and (1.4.5) together are usually called the Hamilton-Jacobi-Bellman equations. In the nonlinear case, we may represent the adjoint variable in the form $\Lambda(t) = 2Qx(t) + \lambda(t)$ and the cost functional in the form $V(x) = x^*Qx + v(x)$, where $\lambda(t)$ and $v(x)$ are the higher order terms, therefore we can have $\Lambda = \frac{\partial V(x)}{\partial x}$.

If we substitute $\frac{\partial V(x)}{\partial x}$ in (1.4.5) for Λ , then the resulting Hamiltonian function is equal to zero, which is the necessary condition (1.2.21) for the optimal control. By Theorem 1, we can also see that (1.2.21) is also the sufficient condition for the optimal control.

In section 1.5 we will make use of (1.4.4) and (1.4.5) to extend the Kleinman-Newton method to the nonlinear-nonquadratic problem.

1.5 Kleinman-Newton Method for Linear Quadratic Problem

The original Kleinman-Newton method reference is a numerical method to compute the optimal control in linear systems with quadratic cost. We briefly describe the method in this section and then in the next section we will describe an extension of this method to solve nonlinear-nonquadratic problems.

Following equations (1.3.9) and (1.3.10), we have

$$(A + BK)^* Q + Q(A + BK) + W = 0. \quad (1.5.1)$$

In [Kleinman] (see also [Russell]), Kleinman shows that by iterating (1.3.9) and (1.5.1) alternatively, or rather by iterating (1.5.2) below, starting with any stabilizing feedback control K_0 ,

$$\begin{cases} K_{i+1} = -U^{-1}(B^* Q_i + R^*) \\ (A + BK_{i+1})^* Q_{i+1} + Q_{i+1}(A + BK_{i+1}) + W = 0 \end{cases} \quad (1.5.2)$$

where the unique positive definite symmetric solution Q of (1.5.1) is found by solving $(A + BK_i)^* Q_i + Q_i(A + BK_i) + W = 0$ using *Smith's method* that may be found in [Russell], for example, for $i = 0, 1, 2, \dots$. We obtain sequences of Q_i and K_i converging quadratically to Q and \hat{K} as $i \rightarrow \infty$.

The individual steps of (1.5.2) can be carried out using *Smith's method* for matrix Liapounov equations or by *Potter's eigenvalue equation method* [Potter]. In brief, *Smith's method* may be described as follows.

For any two stability matrices (i.e. matrices eigenvalues have negative real parts) $A_{m \times m}$ and $B_{n \times n}$, the equation

$$AX + XB = C \quad (1.5.3)$$

can be rewritten into

$$(rI_m - A)X(rI_n - B) - (rI_m + A)X(rI_n + B) = -2rC. \quad (1.5.4)$$

where $r < 0$. Since the eigenvalues of A and B all have negative real parts, both $rI_m + A$ and $rI_n + B$ are nonsingular. Multiplying (1.5.4) by $(rI_m + A)^{-1}$ and $(rI_n + B)^{-1}$ respectively, we have

$$X = (rI_m + A)^{-1}(rI_m - A)X(rI_n - B)(rI_n + B)^{-1} + 2r(rI_m + A)^{-1}C(rI_n + B)^{-1}. \quad (1.5.5)$$

Let

$$U = (rI_m + A)^{-1}(rI_m - A), \quad (1.5.6)$$

$$V = (rI_n - B)(rI_n + B)^{-1}, \quad (1.5.7)$$

$$Y = 2r(rI_m + A)^{-1}C(rI_n + B)^{-1}, \quad (1.5.8)$$

and by putting them into (1.5.5), we can obtain the "fixed point" form

$$X = UXV + Y \quad (1.5.9)$$

which is approximately solved by the iteration procedure

$$X_{j+1} = UX_jV + Y, \quad (1.5.10)$$

where $j = 1, 2, 3, \dots$

We can summarize the main points of Kleinman's method as follows:

- 1) Each Q_i is positive definite symmetric matrix, and each K_i is a stabilizing feedback matrix for the linear quadratic system (1.3.2).
- 2) $\lim_{i \rightarrow \infty} Q_i = Q$, and $\lim_{i \rightarrow \infty} K_i = \hat{K}$.
- 3) The convergence is quadratic.

1.6 Extension of the Kleinman-Newton Method to the Nonlinear Nonquadratic Case

Following equation (2.2.21), we will establish two equations which may be considered to form the basis for the iteration equations we propose to use in solving nonlinear-nonquadratic problems. These equations are expressed in terms of the Hamiltonian (1.2.16):

$$\begin{cases} H(t, \hat{x}(t), u(t), \Lambda(t)) = 0 \\ \frac{\partial H(t, \hat{x}(t), \hat{u}(t), \Lambda(t))}{\partial u} = 0. \end{cases} \quad (1.6.1)$$

From (1.2.16) these equations, in more detail, are

$$\begin{cases} \Lambda^* F(\hat{x}, u) + G(\hat{x}, u) = 0 \\ \Lambda^* \frac{\partial F(\hat{x}, \hat{u})}{\partial u} + \frac{\partial G(\hat{x}, \hat{u})}{\partial u} = 0. \end{cases} \quad (1.6.2)$$

The adjoint variable Λ satisfies the differential equation

$$\frac{d\Lambda^*}{dt} = -\Lambda^* \frac{\partial F(\hat{x}, \hat{u})}{\partial x} - \frac{\partial G(\hat{x}, \hat{u})}{\partial x} \quad (1.6.3)$$

with boundary condition $\lim_{t \rightarrow \infty} \Lambda(t) = 0$. It is clear that for x of dimension more than one, the differential equation (1.6.3) is may be complicated, particularly since it is coupled to the nonlinear state equation (1.4.1). By the previous assumption the adjoint variable $\Lambda(t)$ has the form $2Qx(t) + \lambda(t)$ and the cost function $V(x)$ the form $\hat{u} = K(\hat{x}, \Lambda)$, where $\lambda(t)$ and $v(x)$ are the higher order nonlinear terms, and $\Lambda = \frac{\partial V(x)}{\partial x}$. Substituting Λ in (1.6.2) for $\frac{\partial V(x)}{\partial x}$, we obtain the equations in the same form as those in Theorem 1 and Lemma 1 of Section 1.4.

For the nonlinear-nonquadratic system, Theorem 1 of Section 1.4 gives the necessary and sufficient condition for the optimal control problem. Lemma 1 in Section 1.4 gives the relations among V , $F(x, u)$, and $G(x, u)$. Therefore, by putting together (1.4.1), (1.4.4), and (1.4.5), we construct an iteration relation for numerically solving the optimal control problem in nonlinear-nonquadratic case

$$\left\{ \begin{array}{l} \dot{x}_j = F(x_j, u_j); \\ \frac{\partial V_j(x)}{\partial x} F(x_j, u_j) + G(x_j, u_j) = 0; \\ \frac{\partial V_j(x)}{\partial x_j} \frac{\partial F(x_j, u_{j+1})}{\partial u_{j+1}} + \frac{\partial G(x_j, u_{j+1})}{\partial u_{j+1}} = 0. \end{array} \right. \quad (1.6.4)$$

We will refer to (1.6.4) as the *Kleinman-Newton method* in the nonlinear-nonquadratic case.

Given any stabilizing feedback control $u_0(t) = K_0(x(t))$ with the initial condition $x(0) \in R^n$, we solve $\dot{x} = F(x, u)$ for x in step 0, and then substitute the solution x into $\frac{\partial V(x)}{\partial x} F(x, u) + G(x, u) = 0$ to solve for V_0 . Finally we substitute the solution x and V_0 with the control $u_0(t) = K_0(x(t))$ into $\frac{\partial V(x)}{\partial x} \frac{\partial F(x, u)}{\partial u} + \frac{\partial G(x, u)}{\partial u} = 0$ to solve for $u_1(t) = K_1(x(t))$. In such an iteration, as the step index j goes to infinity, $V_j(x)$ and $u_j(x)$ may be shown to converge respectively to $\hat{V}(x)$ and $\hat{u}(x)$ yielding the lowest cost relative to (1.4.2).

Chapter 2. The Proof of Convergence of the Kleinman-Newton Method

In this chapter we first of all show an equation equivalent to (1.4.5), and then starting with the linear quadratic case, we finally use the contraction mapping method to accomplish the proof.

2.1 The Iteration Formula of the Kleinman-Newton Method

The iteration is described in equation (2.1.1) for $j = 1, 2, 3, \dots$ Starting with a stabilizing control u_0 , the iteration is carried out in three steps in the following order:

$$\left\{ \begin{array}{l} \dot{x}_j = F(x_j, u_j); \\ \frac{\partial V_j(x)}{\partial x} F(x_j, u_j) + G(x_j, u_j) = 0; \\ \frac{\partial V_j(x)}{\partial x_j} \frac{\partial F}{\partial u}(x_j, u_{j+1}) + \frac{\partial G}{\partial u}(x_j, u_{j+1}) = 0. \end{array} \right. \quad (2.1.1)$$

The last equation takes the form shown because

$$\frac{\partial V(x)}{\partial x} F(x, u) = \frac{dV(x(t))}{dt} \quad (2.1.2)$$

and by (2.1.1) and (2.1.2), we have

$$\frac{dV(x(t))}{dt} = -G(x(t), u(t)), \quad (2.1.3)$$

where $G(x, u) = \frac{1}{2}x^*Wx + \frac{1}{2}u^*Uu + g(x, u)$. Therefore we obtain

$$V(x(t)) = \int_t^\infty G(x(s), u(s))ds. \quad (2.1.4)$$

To integrate (2.1.4) is equivalent to solving $\frac{\partial V(x)}{\partial x}F(x, u) + G(x, u) = 0$, and it is clear that it will generally be much easier to integrate (2.1.4) numerically than to solve $\frac{\partial V(x)}{\partial x}F(x, u) + G(x, u) = 0$. Therefore the iteration equation (2.1.1) becomes

$$\left\{ \begin{array}{l} \dot{x}_j = F(x_j, u_j); \\ V_j(x(t)) = \int_t^\infty G(x_j, u_j)ds; \\ \frac{\partial V_j(x)}{\partial x_j} \frac{\partial F}{\partial u}(x_j, u_{j+1}) + \frac{\partial G}{\partial u}(x_j, u_{j+1}) = 0; \end{array} \right. \quad (2.1.5)$$

for $j = 1, 2, 3, \dots$

2.2 The Related Hamiltonian Function and Adjoint Equation

As discussed in Sections 1.1 and 1.6, to find the optimal control u is to minimize the cost $V(X(t)) = \int_t^\infty G(X(s), u(s))ds$, i.e. to minimize the Hamiltonian function

$$H(x, u, \Lambda) = \Lambda^*F(x, u) + G(x, u) \quad (2.2.1)$$

where the adjoint state $\Lambda(t)$ satisfies $\frac{d\Lambda^*}{dt} = -\Lambda^* \frac{\partial F(\hat{x}, \hat{u})}{\partial x} - \frac{\partial G(\hat{x}, \hat{u})}{\partial x}$ with $\lim_{t \rightarrow \infty} \Lambda(t) = 0$ and where $\hat{x}(t)$ and $\hat{u}(t)$ are the optimal trajectory and the optimal control. For each t , minimization with respect to u corresponds to

$$\min_u H(\hat{x}(t), u(t), \Lambda(t)) = 0. \quad (2.2.2)$$

If we assume that $H(x, u, \Lambda)$ is differentiable with respect to u , then the optimal control $\hat{u}(t)$ can be obtained by solving the equation $\frac{\partial H(\hat{x}, \hat{u}, \Lambda)}{\partial u} = 0$ that is

$$\Lambda^* \frac{\partial F(\hat{x}, \hat{u})}{\partial u} + \frac{\partial G(\hat{x}, \hat{u})}{\partial u} = 0. \quad (2.2.3)$$

As shown in [Lukes], the implicit function theorem shows for small values of x , u , and Λ , that (2.2.3) is uniquely solvable for u in terms of x and Λ : $u = K(x, \Lambda)$. It follows that the optimal control can be represented in the form $\hat{u} = K(\hat{x}, \Lambda)$.

For the corresponding linear quadratic problem in which the nonlinear terms $f(x, u)$ and $g(x, u)$ are identically equal to zero, we have

$$\Lambda(t) = 2Q\hat{x}(t) \quad (2.2.4)$$

and

$$V(x(t)) = \hat{x}(t)^* Q \hat{x}(t) \quad (2.2.5)$$

that is

$$\Lambda(t) = \frac{\partial V(x(t))}{\partial x} \quad (2.2.6)$$

where Q is the unique symmetric positive definite solution of the matrix quadratic equation

$$A^* Q + QA + W - QBU^{-1}B^* Q = 0, \quad (2.2.7)$$

and the optimal control is $\hat{u}(t) = -U^{-1}B^* Q \hat{x}(t)$.

For the nonlinear-nonquadratic case, we will assume

$$V(\hat{x}(t)) = \hat{x}(t)^* Q \hat{x}(t) + v(\hat{x}(t)), \quad (2.2.8)$$

and

$$\Lambda(t) = \frac{\partial V(\hat{x}(t))}{\partial x} = 2Q\hat{x}(t) + \lambda(t). \quad (2.2.9)$$

Therefore the optimal control problem (2.1.5) can be equivalently described by solving the iteration equations

$$\begin{cases} \dot{x}_j = F(x_j, u_j), & x_j(0) = x_0, \\ \Lambda_j(t) \frac{\partial F}{\partial u}(x_j, u_{j+1}) + \frac{\partial G}{\partial u}(x_j, u_{j+1}) = 0, & \lim_{t \rightarrow \infty} \Lambda_j(t) = 0. \end{cases} \quad (2.2.10)$$

By (2.2.9) the optimal control can be written as

$$\hat{u} = K(\hat{x}, Q\hat{x}(t) + \lambda(t)) = -U^{-1}B^*(Q\hat{x}(t) + \lambda(t)) + k(\hat{x}(t), \lambda(t)), \quad (2.2.11)$$

and the system equation becomes

$$\begin{aligned} \dot{\hat{x}} &= F(\hat{x}, \hat{u}) \\ &= (A - BU^{-1}B^*Q)\hat{x} - BU^{-1}B^*\lambda + Bk(\hat{x}, \lambda) + f(\hat{x}, k(\hat{x}, Q\hat{x} + \lambda)) \\ &= (A - BU^{-1}B^*Q)\hat{x} + \varphi(\hat{x}, \lambda), \end{aligned} \quad (2.2.12)$$

where $\varphi(\hat{x}, \lambda) = -BU^{-1}B^*\lambda + Bk(\hat{x}, \lambda) + f(\hat{x}, k(\hat{x}, Q\hat{x} + \lambda))$, and is such that $\varphi(0,0) = 0$,

$$\frac{\partial \varphi(0,0)}{\partial x} = 0, \text{ and } \frac{\partial \varphi(0,0)}{\partial \lambda} = 0.$$

The adjoint equation $\frac{d\Lambda^*}{dt} = -\Lambda^* \frac{\partial F(\hat{x}, \hat{u})}{\partial x} - \frac{\partial G(\hat{x}, \hat{u})}{\partial x}$ can be rewritten as

$$\frac{d\Lambda^*}{dt} = -\Lambda^* \left(A + \frac{\partial f(\hat{x}, \hat{u})}{\partial x} \right) - \hat{x}^* W - \frac{\partial g(\hat{x}, \hat{u})}{\partial x}. \quad (2.2.13)$$

By putting (2.2.9) into (2.2.13) we have

$$\frac{d\lambda^*}{dt} = -\lambda^*(A - BU^{-1}B^*Q) - (Q\hat{x} + \lambda)^* \frac{\partial f(\hat{x}, K(\hat{x}, Q\hat{x}) + \lambda)}{\partial x} - \frac{\partial g(\hat{x}, K(\hat{x}, Q\hat{x}) + \lambda)}{\partial x} - \varphi(\hat{x}, \lambda)^* Q \quad (2.2.14)$$

which can be rewritten more concisely in the form

$$\frac{d\lambda^*}{dt} = -(A - BU^{-1}B^*Q)\lambda^* - \psi(\hat{x}, \lambda), \quad (2.2.15)$$

where $\psi(\hat{x}, \lambda) = -(Q\hat{x} + \lambda)^* \frac{\partial f(\hat{x}, K(\hat{x}, Q\hat{x}) + \lambda)}{\partial x} - \frac{\partial g(\hat{x}, K(\hat{x}, Q\hat{x}) + \lambda)}{\partial x} - \varphi(\hat{x}, \lambda)^* Q$ and

where $\psi(x, \lambda)$ is such that $\psi(0,0) = 0$, $\frac{\partial \psi(0,0)}{\partial x} = 0$, and $\frac{\partial \psi(0,0)}{\partial \lambda} = 0$.

Consequently the iteration (2.2.10) can be written in the equivalent form

$$\begin{cases} \dot{\hat{x}} = (A - BU^{-1}B^*Q)\hat{x} + \varphi(\hat{x}, \lambda); \\ \frac{d\lambda^*}{dt} = -(A - BU^{-1}B^*Q)\lambda^* - \psi(\hat{x}, \lambda). \end{cases} \quad (2.2.16)$$

Then, using the boundary conditions at 0 and ∞ we can integrate to obtain equivalent integral equations

$$\begin{cases} x_j = e^{(A - BU^{-1}B^*Q)t} x_0 + \int_0^t e^{(A - BU^{-1}B^*Q)(t-s)} \varphi(x_j(s), \lambda_j(x_j(s))) ds; \\ \lambda_{j+1}(t) = -\int_t^\infty e^{(A - BU^{-1}B^*Q)^*(s-t)} \psi(x_j(s), \lambda_j(x_j(s))) ds. \end{cases} \quad (2.2.17)$$

The new control generated at each iteration stage is

$$u_{j+1} = K_{j+1}(\hat{x}, Q\hat{x}(t) + q_j(x)) = -U^{-1}B^*(Qx(t) + q_j(t)) + k_{j+1}(x(t), q_j(x)), \quad (2.2.18)$$

where $q(x)$ is a function with $q(0) = 0$, and $\frac{\partial q(0)}{\partial x} = 0$ such that $\lambda(0) = q(x(0)) = q(x_0)$

and $q_{j+1}(x_0) = \lambda_{j+1}(0)$. Hence we can see that proving the convergence of (2.1.5) is equivalent to proving the convergence of (2.2.17).

2.3 The Proof of the Convergence of (2.2.17)

For convenience, let $A_Q = A - BU^{-1}B^*Q$ be the cost matrix derived from the linear-quadratic theory, which is shown by (2.2.7) and (2.2.11). Then, (2.2.17) can be written as

$$x = e^{A_Q t} x_0 + \int_0^t e^{A_Q(t-s)} \varphi(x(s), \lambda(s)) ds, \quad (2.3.1)$$

$$\lambda(t) = -\int_t^\infty e^{A_Q^*(s-t)} \psi(x(s), \lambda(s)) ds. \quad (2.3.2)$$

Since A_Q is a stability matrix, there are positive numbers R and α such that, for $t \geq 0$,

$$\|e^{A_Q t}\| = \|e^{A_Q^* t}\| \leq R e^{-\alpha t}. \quad (2.3.3)$$

In the region of interest, we have assumed that

$$\begin{cases} \varphi(0,0) = 0, & \frac{\partial \varphi(0,0)}{\partial x} = 0, & \frac{\partial \varphi(0,0)}{\partial \lambda}; \\ \psi(0,0) = 0, & \frac{\partial \psi(0,0)}{\partial x} = 0, & \frac{\partial \psi(0,0)}{\partial \lambda}. \end{cases} \quad (2.3.4)$$

hence we can further assume that

$$\|\varphi(x_1, \lambda_1) - \varphi(x_2, \lambda_2)\| \leq C(\|x_1\| + \|x_2\| + \|\lambda_1\| + \|\lambda_2\|)(\|x_1 - x_2\| + \|\lambda_1 - \lambda_2\|), \quad (2.3.5)$$

$$\|\psi(x_1, \lambda_1) - \psi(x_2, \lambda_2)\| \leq C(\|x_1\| + \|x_2\| + \|\lambda_1\| + \|\lambda_2\|)(\|x_1 - x_2\| + \|\lambda_1 - \lambda_2\|), \quad (2.3.6)$$

for some constant C . Setting $x = x_1$, $\lambda = \lambda_1$, $x_2 = 0$, $\lambda_2 = 0$, we also have

$$\|\varphi(x, \lambda)\| \leq C(\|x\| + \|\lambda\|)^2, \quad (2.3.7)$$

$$\|\psi(x, \lambda)\| \leq C(\|x\| + \|\lambda\|)^2. \quad (2.3.8)$$

From (2.2.7) we know that Q and A_Q together satisfy a Lyapounov matrix

equation of the form

$$A_Q^* Q + QA_Q + W = 0, \quad (2.3.9)$$

for some positive definite symmetric matrix W . We know that the ellipsoids in R^n given by

$$E_r = \{x \mid x^* Qx \leq r^2\} \quad (2.3.10)$$

are invariant under the linear system $\dot{x} = A_Q x$, because

$$\frac{d(x^* Qx)}{dt} = \dot{x}^* Qx + x^* Q\dot{x} = x^* A^* Qx + x^* QA x = -x^* Wx \leq 0. \quad (2.3.11)$$

Since Q is a positive definite symmetric matrix, letting Λ be its diagonal eigenvalue matrix, we have $Qx = P^* \Lambda Px$, where P is an orthogonal matrix. Making this substitution in (2.3.10) we have $x^* Qx = (Px)^* \Lambda Px \leq r^2$. Letting q^2 be the reciprocal of the smallest eigenvalue of Q , (i.e., the smallest diagonal entry of Λ^{-1}) we have

$$\|x\| = \|Px\| \leq qr, \quad x \in E_r. \quad (2.3.12)$$

Our object is to show the existence of a function $\lambda(x)$ such that equation (2.3.1) and (2.3.2) are satisfied and to show that $\lambda(x)$ is obtained as the limit of the iteration process described above. To this end we fix a positive value M and consider the metric space $\mathcal{M}_{r,M}$ consisting of function $\lambda(x)$ defined on E_r with the properties

$$\lambda(0) = 0, \quad \|\lambda(x) - \lambda(y)\| \leq m\|x - y\|, \quad x, y \in E_r, \quad (2.3.13)$$

for some positive number $m \leq M$. We define $\|\lambda\|_m$ to be the greatest lower bound of the set of m for which (2.3.13) holds. It is easy to verify that $\|\lambda\|_m$, as just defined, is a norm and that, with the corresponding distance function, $\mathcal{M}_{r,M}$ is a complete metric space. Combining the two parts of (2.3.13) with this definition we have

$$\|\lambda(x) - \lambda(y)\| \leq \|\lambda\|_m \|x - y\|, \quad \|\lambda(x)\| \leq \|\lambda\|_m \|x\|, \quad x, y \in E_r. \quad (2.3.14)$$

Further, for M as just fixed, a standard argument using x^*Qx as a Lyapounov function shows that there is a positive number r_M such that the ellipsoid E_r remains invariant under the nonlinear system

$$\dot{x} = A_Q x + \varphi(x, \lambda(x)), \quad (2.3.15)$$

for $r \leq r_M$. The argument can be simply stated as

$$\begin{aligned} \frac{d(x^*Qx)}{dt} &= \dot{x}^*Qx + x^*Q\dot{x} \\ &= x^*A^*Qx + x^*QAx + \varphi^*Qx + x^*Q\varphi \\ &= -x^*Wx + \varphi^*Qx + x^*Q\varphi \leq 0, \end{aligned} \quad (2.3.16)$$

when r_M is sufficiently small, $\|\varphi^*Qx + x^*Q\varphi\| \leq \|x^*Wx\|$ for $x \in E_r$. We assume henceforth that r is restricted in this way, and so that (2.3.12) holds in the nonlinear case.

For a given function $\lambda(x) \in \mathcal{M}_{r,M}$ we construct trajectories $x(t)$ lying in E_r via (2.3.1); these are solutions of (2.3.15) having the property $\lim_{t \rightarrow \infty} \|x(t)\| = 0$. We employ the standard contraction fixed point approach to establish the existence of $\lambda(x)$ such that (2.3.1) and (2.3.2) are satisfied. Thus for given $\lambda(x)$ we construct trajectories $x(t)$ from (2.3.1), then we determine a new function $\tilde{\lambda}(x)$ by setting (cf. (2.3.2))

$$\tilde{\lambda}(x(t)) = -\int_t^\infty e^{A_Q^*(s-t)} \psi(x(s), \lambda(x(s))) ds. \quad (2.3.17)$$

This provides a mapping from the function $\lambda(x)$ to the function $\tilde{\lambda}(x)$. We need to show that this mapping leaves $\mathcal{M}_{r,M}$ invariant, i.e., carries that space into itself, and that the mapping is a contraction. For this we will need a series of estimates which now follow.

First of all we obtain an estimate on the trajectories. We suppose that $x(t)$ and $y(t)$ are both solutions of (2.3.1), for the same $\lambda(x) \in \mathcal{M}_{r,M}$, corresponding to initial points x_0 and y_0 in E_r , respectively. Clearly we have

$$x(t) - y(t) = e^{A_\rho t} (x_0 - y_0) + \int_0^t e^{A_\rho(t-s)} [\varphi(x(s), \lambda(x(s))) - \varphi(x(s), \lambda(y(s)))] ds. \quad (2.3.18)$$

From (2.3.3) we have

$$\|e^{A_\rho t} (x_0 - y_0)\| = R e^{-\alpha t} \|x_0 - y_0\|, \quad t \geq 0. \quad (2.3.19)$$

On the other hand, using (2.3.3) again together with (2.3.5), (2.3.7), and (2.3.14), we see that the norm of the integral is bounded by

$$\begin{aligned} & \int_0^t R e^{-\alpha(t-s)} C [\|x(s)\| + \|y(s)\| + \|\lambda(x(s))\| + \|\lambda(y(s))\|] [\|x(s) - y(s)\| + \|\lambda(x(s)) - \lambda(y(s))\|] ds \\ & \leq \int_0^t R e^{-\alpha(t-s)} C (1+M)^2 (\|x(s)\| + \|y(s)\|) \|x(s) - y(s)\| ds. \end{aligned} \quad (2.3.20)$$

Since the existence of a unique solution of (2.3.1) has been proved by [Cronin] by taking a convergent sequence arising out of the fixed point procedure, we can assume an estimate of the form

$$\|x(t) - y(t)\| \leq \rho e^{-\alpha t} \|x_0 - y_0\|, \quad t \geq 0, \quad (2.3.21)$$

holds, and we note that this includes

$$\|x(t)\| \leq \rho e^{-\alpha t} \|x_0\|, \quad t \geq 0, \quad (2.3.22)$$

as a consequence of setting $y_0 = 0$ (hence $y(t) \equiv 0$), then this estimate remains valid using (2.3.14), (2.3.21), and (2.3.22) under (2.3.18) provided that

$$\rho e^{-\alpha t} \|x_0 - y_0\| \geq R e^{-\alpha t} \|x_0 - y_0\| + \int_0^t R e^{-\alpha(t-s)} C (1+M)^2 \rho^2 e^{-2\alpha s} (\|x_0\| + \|y_0\|) \|x_0 - y_0\| ds. \quad (2.3.23)$$

By further restricting r if necessary, and by (2.3.12), we can ensure

$$1 \geq \frac{1}{2} \left[1 + \int_0^t e^{-\alpha s} C(1+M)^2 4R^2 2qr ds \right] = \frac{1}{2} + \frac{4C(1+M)^2 R^2 qr}{\alpha}, \quad (2.3.24)$$

and then (2.3.23) holds for the value $\rho = 2R$. So now we assume

$$\|x(t) - y(t)\| \leq 2R e^{-\alpha t} \|x_0 - y_0\|, \quad t \geq 0. \quad (2.3.25)$$

The next step is to estimate $\|x_\lambda(s) - x_\mu(s)\|$ in terms of $\|\lambda - \mu\|_m$ for trajectories $x_\lambda(t)$, $x_\mu(t)$ corresponding to $\lambda(x)$, $\mu(x)$, respectively, in $\mathcal{M}_{r,M}$, and corresponding to the same initial point $x_0 \in E_r$. In this case we have

$$x_\lambda(t) - x_\mu(t) = \int_0^t e^{A_\varphi(t-s)} \left[\varphi(x_\lambda(s), \lambda(x_\lambda(s))) - \varphi(x_\mu(s), \mu(x_\mu(s))) \right] ds. \quad (2.3.26)$$

Again, the norm of the right hand is bounded by

$$\int_0^t R e^{-\alpha(t-s)} C \left[\|x_\lambda(s)\| + \|x_\mu(s)\| + \|\lambda(x_\lambda(s))\| + \|\mu(x_\mu(s))\| \right] \left[\|x_\lambda(s) - x_\mu(s)\| + \|\lambda(x_\lambda(s)) - \mu(x_\mu(s))\| \right] ds.$$

Using (2.3.14), (2.3.21), (2.3.22), and by taking $\rho = 2R$, we have

$$\|x_\lambda(s)\| + \|x_\mu(s)\| + \|\lambda(x_\lambda(s))\| + \|\mu(x_\mu(s))\| \leq 4(1+M)R e^{-\alpha s} \|x_0\|, \quad (2.3.27)$$

and

$$\begin{aligned} & \|x_\lambda(s) - x_\mu(s)\| + \|\lambda(x_\lambda(s)) - \mu(x_\mu(s))\| \\ &= \|x_\lambda(s) - x_\mu(s)\| + \|\lambda(x_\lambda(s)) - \lambda(x_\mu(s)) + \lambda(x_\mu(s)) - \mu(x_\mu(s))\| \\ &\leq \|x_\lambda(s) - x_\mu(s)\| + \|\lambda(x_\lambda(s)) - \lambda(x_\mu(s))\| + \|\lambda(x_\mu(s)) - \mu(x_\mu(s))\| \\ &\leq \|x_\lambda(s) - x_\mu(s)\| + \|\lambda\|_m \|x_\lambda(s) - x_\mu(s)\| + \|\lambda - \mu\|_m \|x_\mu(s)\| \\ &= (1 + \|\lambda\|_m) \|x_\lambda(s) - x_\mu(s)\| + \|\lambda - \mu\|_m \|x_\mu(s)\| \\ &\leq (1+M) \|x_\lambda(s) - x_\mu(s)\| + \|\lambda - \mu\|_m 2R e^{-\alpha s} \|x_0\|, \end{aligned} \quad (2.3.28)$$

then by (2.3.27) and (2.3.28), the right hand of (2.3.26) is bounded by

$$e^{-\alpha t} \int_0^\infty 4CR^2(1+M) \|x_0\| \left[(1+M) \|x_\lambda(s) - x_\mu(s)\| + \|\lambda - \mu\|_m 2R e^{-\alpha s} \|x_0\| \right] ds. \quad (2.3.29)$$

Similar to (2.3.21), we anticipate an estimate of the form

$$\|x_\lambda(t) - x_\mu(t)\| \leq N e^{-\alpha t} \|\lambda - \mu\|_m \|x_0\|, \quad (2.3.30)$$

and substitute it into the expression of (2.3.29) to obtain

$$e^{-\alpha t} \|\lambda - \mu\|_m \int_0^\infty 4CR^2(1+M) \|x_0\|^2 e^{-\alpha s} [(1+M)N + 2R] ds. \quad (2.3.31)$$

Accept $N = R$ and we see that we need

$$e^{-\alpha t} \|\lambda - \mu\|_m \int_0^\infty 4CR^2(1+M) \|x_0\|^2 e^{-\alpha s} (3+M)R ds \leq R e^{-\alpha t} \|\lambda - \mu\|_m \|x_0\|, \quad (2.3.32)$$

which, since $\|x_0\| \leq qr$, is satisfied if

$$\frac{4CR^2(1+M)(3+M)qr}{\alpha} \leq 1. \quad (2.3.33)$$

Since we can assure $N = R$ by further restricting R if necessary, we may now assume that (2.3.30) is valid with $N = R$, i.e.,

$$\|x_\lambda(t) - x_\mu(t)\| \leq R e^{-\alpha t} \|\lambda - \mu\|_m \|x_0\|. \quad (2.3.34)$$

The final step is to start with two functions $\lambda(x)$, $\mu(x)$ in $\mathcal{M}_{r,M}$. Under the mapping (2.3.17) we then obtain two new functions $\tilde{\lambda}(x)$, $\tilde{\mu}(x)$, respectively, and we have, setting $t = 0$,

$$\tilde{\lambda}(x_0) - \tilde{\mu}(x_0) = \tilde{\lambda}(x_\lambda(0)) - \tilde{\mu}(x_\mu(0)) = - \int_0^\infty e^{A_\psi^* s} \left[\psi(x_\lambda(s), \lambda(x_\lambda(s))) - \psi(x_\mu(s), \mu(x_\mu(s))) \right] ds. \quad (2.3.35)$$

Using (2.3.27) and (2.3.28), we can bound the integral here by

$$\begin{aligned} & \int_0^\infty R e^{-\alpha s} C \left[\|x_\lambda(s)\| + \|x_\mu(s)\| + \|\lambda(x_\lambda(s))\| + \|\mu(x_\mu(s))\| \right] \left[\|x_\lambda(s) - x_\mu(s)\| - \|\lambda(x_\lambda(s)) - \mu(x_\mu(s))\| \right] \\ & \leq \int_0^\infty R^2 e^{-2\alpha s} 4C(1+M) \|x_0\| \left[(1+M) \|x_\lambda(s) - x_\mu(s)\| - \|\lambda - \mu\| \|x_\mu(s)\| \right] ds. \end{aligned} \quad (2.3.36)$$

Then using (2.3.34) and (2.3.35) with the selected value $\rho = 2R$ and $\|x_0\| \leq qr$, we have

$$\|\tilde{\lambda}(x_0) - \tilde{\mu}(x_0)\| \leq -\|\lambda - \mu\|_m \|x_0\| \int_0^\infty R^3 e^{-3\alpha s} 4C(1+M)(3+M)qr ds, \quad (2.3.37)$$

and it is clear that, given γ with $0 < \gamma < 1$, by further restricting r , if necessary, we can obtain

$$\|\tilde{\lambda}(x_0) - \tilde{\mu}(x_0)\| \leq \gamma \|\lambda - \mu\|_m \|x_0\|. \quad (2.3.38)$$

Since this is true for all $x_0 \in E_r$, we have

$$\|\tilde{\lambda} - \tilde{\mu}\|_m \leq \gamma \|\lambda - \mu\|_m. \quad (2.3.39)$$

An entirely similar argument, involving just $\lambda(x_0)$ instead of the difference $\lambda(x_0) - \mu(x_0)$ also shows that with r suitably restricted we have

$$\|\tilde{\lambda}\|_m \leq \|\lambda\|_m \leq M, \quad (2.3.40)$$

for $\lambda \in \mathcal{M}_{r,M}$. Combining (2.3.39) and (2.3.40) we see that the map (2.3.17) leaves $\mathcal{M}_{r,M}$ invariant and is a contraction with respect to the distance function $\|\lambda - \mu\|_m$. It then follows from the contraction fixed point theorem that there is a unique function $\lambda \in \mathcal{M}_{r,M}$ such that the equation (2.3.1) and (2.3.2) hold.

However, the existence of $\lambda(x)$ is something known already, from the paper of Lukes, e.g. . The significance of the present result lies in the fact that the iteration procedure which corresponds to application of the Contraction Fixed Point Theorem in the present instance coincides with to the Kleinman-Newton iteration scheme. That is, given an initial function $\lambda_1(x)$ in $\mathcal{M}_{r,M}$, one first defines the solution $x_1(t)$ with initial state x_0 , as the unique solution of the integral equation

$$x_1 = e^{A_Q t} x_0 + \int_0^t e^{A_Q(t-s)} \varphi(x_1(s), \lambda(x_1(s))) ds. \quad (2.3.41)$$

This gives a family of trajectories covering the whole ellipsoidal neighborhood E_r of the origin in R^n . Then one defines a new function $\lambda_2(x)$ by integrating over each of the trajectories defined by (2.3.41):

$$\lambda_2(x_1(t)) = - \int_t^\infty e^{A_Q^*(s-t)} \psi(x_1(s), \lambda_1(x_1(s))) ds. \quad (2.3.42)$$

This corresponds, in fact, to integrating the cost function corresponding to the control determined by $\lambda_1(x)$ over the trajectories $x_1(t)$ to obtain the new cost function $V_2(x)$, whose gradient is then the vector function $2Qx + \lambda_2(x)$. Later steps, of course, take the form of (2.2.17), i.e.,

$$\begin{cases} x_j = e^{(A-BU^{-1}B^*Q)t} x_0 + \int_0^t e^{(A-BU^{-1}B^*Q)(t-s)} \varphi(x_j(s), \lambda_j(x_j(s))) ds; \\ \lambda_{j+1}(t) = - \int_t^\infty e^{(A-BU^{-1}B^*Q)^*(s-t)} \psi(x_j(s), \lambda_j(x_j(s))) ds. \end{cases} \quad (2.3.43)$$

In each iteration step, the gradient of the new cost function $V_{j+1}(x)$ is the vector function $2Qx + \lambda_{j+1}(x)$.

Chapter 3. Using the Kleinman-Newton Method to Compute the Optimal Control

In this chapter we will describe in detail the numerical algorithm we have devised for implementing the Kleinman-Newton method. A two dimensional control system is used to demonstrate the success of the method. The convergence rate of the method seems to be quadratic. Some notations and indices may be different from the first two chapters; they will be clearly defined in the text to follow.

3.1 Assumptions

The system is given as

$$\dot{x} = F(x, u), \text{ with } x(0) = x_0 \in R^n \text{ and } u_0 = K_0(x) \in R^m, \quad (3.1.1)$$

where $F(x, u) = Ax(t) + Bu(x(t)) + f(x, u)$, $x \in R^n$, $u \in R^m$, x_0 is the initial condition, and u_0 is the initial control which is chosen to make the system asymptotically stable. Since the method works only within a local area around origin, x_0 has to be restricted in a domain $D \subset R^n$ in which a corresponding initial stabilizing control u_0 can be found. Moreover, we will let vector $(\alpha_1 \ \alpha_2 \ \dots \ \alpha_n)^T$ denote x , $(\mu_1 \ \mu_2 \ \dots \ \mu_m)^T$ denote control u , and $(f_1 \ f_2 \ \dots \ f_n)^T$ denote function $F(x, u)$.

As we have discussed in Chapter 2 and Section 2.1, we will use the following iteration equations modified from (2.1.5) in stead of directly using the Hamilton-Jacobi-Bellman equations (2.1.1)

$$\begin{cases} \dot{x}_j = F(x_j, u_j); \\ V_j(x(t)) = \int_t^\infty G(x_j(s), u_j(x(s))) ds; \\ \frac{\partial V_j}{\partial x_j} \frac{\partial F(x_j, u_{j+1})}{\partial u_{j+1}} + \frac{\partial G(x_j, u_{j+1})}{\partial u_{j+1}} = 0. \end{cases} \quad (3.1.2)$$

where the cost function $G(x, u)$ can be defined in several ways. For example, as in the case at the case of pendulum, $G(x, u)$ can be defined as

$$G(x, u) = \sum_{i=1}^n \alpha_i^2 + \sum_{j=1}^m \mu_j^2, \quad (3.1.3)$$

without higher order terms. In this chapter, we will use $G(x, u)$ in this form. Thus the cost is quadratic but the system is nonlinear.

By (3.1.3), we have

$$\frac{\partial G(x, u)}{\partial u} = \left(\frac{\partial G}{\partial \mu_1} \quad \frac{\partial G}{\partial \mu_2} \quad \cdots \quad \frac{\partial G}{\partial \mu_m} \right) = 2(\mu_1 \quad \mu_2 \quad \cdots \quad \mu_m), \quad (3.1.4)$$

hence by $\frac{\partial V}{\partial x} \frac{\partial F(x, u)}{\partial u} + \frac{\partial G(x, u)}{\partial u} = 0$ we have

$$(\mu_1 \quad \mu_2 \quad \cdots \quad \mu_m) = -\frac{1}{2} \frac{\partial V}{\partial x} \frac{\partial F(x, u)}{\partial u}$$

$$= -\frac{1}{2} \begin{pmatrix} \frac{\partial V}{\partial \alpha_1} & \frac{\partial V}{\partial \alpha_2} & \cdots & \frac{\partial V}{\partial \alpha_n} \end{pmatrix} \begin{pmatrix} \frac{\partial f_1}{\partial \mu_1} & \frac{\partial f_1}{\partial \mu_2} & \cdots & \frac{\partial f_1}{\partial \mu_m} \\ \vdots & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ \frac{\partial f_n}{\partial \mu_1} & \frac{\partial f_n}{\partial \mu_2} & \cdots & \frac{\partial f_n}{\partial \mu_m} \end{pmatrix}. \quad (3.1.5)$$

There are 5 crucial points we need to pay attention to:

- 1) Since the Kleinman-Newton method is a numerical method, we must first define a fixed domain D for x and then partition domain D into a finite number of elements whose vertices are called grid points at which the numerical values of u and V will be found (in two dimensional space the element can be a square or rectangle, and in three dimensional space, a cubic, etc.).
- 2) Because the feedback control u is a function of x , before the numerical iterations in solving (3.1.1), the values of u at all the grid points must be found in the domain D ; and in solving the value of trajectory $x(t)$ in each iteration step
 - If the trajectory $x(t)$ happens to be at a grid point, then the corresponding control value $u(x)$ at this grid point must be found out from a database which stores the values of $u(x)$ at all the grid points.
 - If the trajectory $x(t)$ happen to be inside of a finite element, the corresponding control value $u(x)$ must be calculated by using interpolation method (cubic spline method is recommended).
- 3) We know that x is a function of t , u is a function of x and at last the function of t too, and trajectory x will become asymptotically stable at origin, so that each value of V can be found by integrating $G(x, u)$ backwards with respect to t (see Figure 3.1).

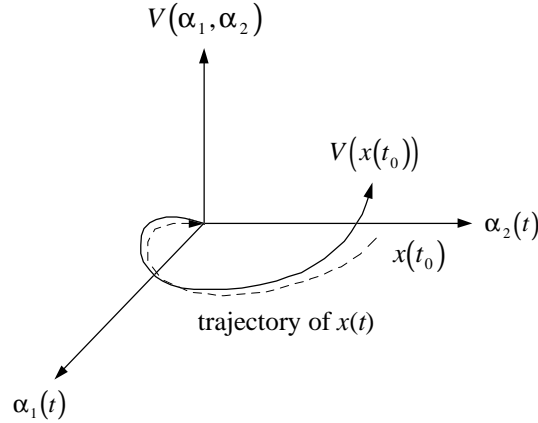


Figure 3.1.1: Each value of $V(x(t_i))$ is found by integrating backwards along the trajectory $x(t)$ with the initial value $x(t_i)$.

- 3) Since V is a function of x in the domain D , in order to find $\frac{\partial V}{\partial x}$ in solving (3.1.2), the values of V at all the grid points must be found by integrating backwards along the trajectory $x(t)$.
- 4) In domain D , the values of x at all the grid points will be used as an initial value in order to find all the necessary values of V by integration (see Figure 3.1), and these initial values remain the same in iterations.
- 5) The values of u , V , G , and F are all numerically discrete, we will use finite sets rather than continuous functions to represent them.

3.2 Computational Algorithm for the Kleinman-Newton Method

Algorithm Step I. Partition the Domain D into Finite Number of Elements

We partition the domain D into finite number of equal sized elements which are easier to program than non-equal sized elements. We define

$$D = \left\{ (d_1 \ d_2 \ \dots \ d_n)^T : d_i \in [-l_i, l_i], 1 \leq i \leq n, l_i \in R^+, i, n \in N \right\}. \quad (3.2.1)$$

Since the finite elements are of the same size, let h be the grid length and let $m_i = \frac{2l_i}{h}$, then there are $m_i + 1$ grid points along each axis. Therefore there will be totally $M = \prod_{i=1}^n (m_i + 1)$ grid points in the domain D which is also the number of initial values of x for the equation (3.1.2), and they are

$$\begin{pmatrix} -l_1 \\ -l_2 \\ \vdots \\ -l_n \end{pmatrix}, \begin{pmatrix} -l_1 + h \\ -l_2 \\ \vdots \\ -l_n \end{pmatrix}, \dots, \begin{pmatrix} -l_1 + m_1 h = l_1 \\ -l_2 \\ \vdots \\ -l_n \end{pmatrix}, \dots, \begin{pmatrix} -l_1 \\ -l_2 + h \\ \vdots \\ -l_n \end{pmatrix}, \begin{pmatrix} -l_1 \\ -l_2 + 2h \\ \vdots \\ -l_n \end{pmatrix}, \dots, \begin{pmatrix} -l_1 \\ -l_2 + m_2 h = l_2 \\ \vdots \\ -l_n \end{pmatrix}, \dots, \\ \begin{pmatrix} l_1 \\ l_2 \\ \vdots \\ -l_n \end{pmatrix}, \begin{pmatrix} l_1 \\ l_2 \\ \vdots \\ -l_n + h \end{pmatrix}, \dots, \begin{pmatrix} l_1 \\ l_2 \\ \vdots \\ -l_n + m_n h = l_n \end{pmatrix}, \text{ where } \begin{cases} -l_i + m_i h = l_i \\ m_i = \frac{2l_i}{h} \end{cases}, \text{ with } i = 1, 2, 3, \dots, n.$$

These grid points are used as initial values in the computation. We denote these initial values as $x_0 = \{x_0^1, x_0^2, \dots, x_0^j, \dots, x_0^M\}$ where $0 \leq j \leq M$ and $M = \prod_{i=1}^n (m_i + 1)$. The values of control u at all the grid points will be denoted as $u_k(x) = \{u_k^1, u_k^2, \dots, u_k^j, \dots, u_k^M\}$, where the subscript k represents the iteration step number. Similarly the values of V at all the grid points will also be denoted as $V_k(x) = \{V_k^1, V_k^2, \dots, V_k^j, \dots, V_k^M\}$.

Algorithm Step II. Initial Stabilizing Control

Each initial value $x_0^j \in \{x_0^1, x_0^2, \dots, x_0^M\}$ corresponds to a stabilizing control value u_0^j , so that M such controls $\{u_0^1, u_0^2, \dots, u_0^j, \dots, u_0^M\}$ must be found at the beginning.

The subscript 0 of x_0^j refers to initial condition and it remains unchanged, whereas the subscript 0 of u_0^j refers to iteration step 0 and it will change to 1, 2, 3, ... until iteration is finished. The superscript of both x_0^j and u_0^j indicates the location of the initial value in domain D . There are several ways to find these initial stabilizing controls. One common way is by guessing. A more systematic way that we recommend here is as follows:

- 1) Linearize $\dot{x} = F(x, u)$ to be $\dot{x} = Ax + Bu$.
- 2) Choose two positive numbers r and λ such that $-(A + \lambda I)$ is a stability matrix which means all of its eigenvalues have negative real part, and $\lambda \geq r$.
- 3) Use Smith's method introduced in Chapter 1 section 1.5 to solve the equation (3.2.2) for the unique positive definite matrix Z .

$$-(A + \lambda I)Z - Z(A + \lambda I)^* + BB^* = 0 \quad (3.2.2)$$

- 4) The desired initial control is $u_0(x) = Kx = -B^*Z^{-1}x$.

Note: The initial control is guaranteed to work only in the domain D , and from next iteration on, the control will be generated by Kleinman-Newton method (3.1.2) rather by (3.2.2).

Algorithm Step III. Start the Iteration

For greater generality, we start with iteration step $k \geq 0$. With each initial value $x_0^i \in \{x_0^1, x_0^2, \dots, x_0^j, \dots, x_0^M\}$ and the corresponding control $u_k(x)$ where k is the iteration step number starting from 0, we solve for $\dot{x} = F(x, u)$ numerically (by Runge Kutta method for instance) to obtain M different trajectories of x from these M initial values towards origin.

We integrate $V_k(x_0^j) = \int_s^{t_f} G(x(s), u_k(x(s))) ds$ with initial value x_0^j along the corresponding trajectory and with t_f large enough to approximate ∞ , so that we can have M values of $V_k^i \in \{V_k^1 \ V_k^2 \ \dots \ V_k^j \ \dots \ V_k^M\}$. Again, the subscript of V_k^j refers to iteration step k start from 0 and it is consistent with the subscript of control u_k , and the superscript of V_k^j denotes location in domain D .

At iteration step k , once we have all the values of $\{V_k^1 \ V_k^2 \ \dots \ V_k^j \ \dots \ V_k^M\}$ at all the grid points, by finite difference method we can calculate one by one each component $\frac{\partial V_k}{\partial \alpha_l}$ of $\frac{\partial V_k}{\partial x}$ where $x = (\alpha_1, \alpha_2, \dots, \alpha_l, \dots, \alpha_n)^T$, $1 \leq j \leq M$, and $1 \leq l \leq n$.

Similarly, we are also able to compute the values of $F(x, u) = (f_1 \ f_2 \ \dots \ f_n)^T$ at all the grid points and use finite difference method to calculate each component $\frac{\partial f_i}{\partial \mu_l}$ of

$\frac{\partial F(x, u)}{\partial u}$, where $u = (\mu_1, \mu_2, \dots, \mu_l, \dots, \mu_m)^T$, $1 \leq i \leq n$ and $1 \leq l \leq m$. Finally by (3.1.5) we

generate all the values of control u_k for the iteration step k . We denote all the values of u_k by $\{u_k^1, u_k^2, \dots, u_k^j, \dots, u_k^M\}$.

Algorithm Step IV. Conclusion of the Iteration

With the values of $\{u_k^1, u_k^2, \dots, u_k^j, \dots, u_k^M\}$ and the initial values $\{x_0^1, x_0^2, \dots, x_0^j, \dots, x_0^M\}$, we start iteration step $k + 1$. By repeating algorithm step III and IV, we will have all the control values that converge to optimal control values $\hat{u}(x) = \{\hat{u}^1, \hat{u}^2, \dots, \hat{u}^j, \dots, \hat{u}^M\}$, and have all the values of V that converge to the minimum ones $\hat{V}(x) = \{\hat{V}^1, \hat{V}^2, \dots, \hat{V}^j, \dots, \hat{V}^M\}$.

3.3 Computational Examples

3.3.1 Pendulum System

We will try a two dimensional system by the Kleinman-Newton method. A pendulum system can be modeled as a second order differential equation

$$\begin{pmatrix} \dot{\alpha}_1 \\ \dot{\alpha}_2 \end{pmatrix} = \begin{pmatrix} \alpha_2 \\ -a \sin \alpha_1 + \mu \end{pmatrix}, \quad (3.3.1)$$

where the constant $a = \frac{6.1799}{2}$, $x = \begin{pmatrix} \alpha_1 \\ \alpha_2 \end{pmatrix}$, and $u = \begin{pmatrix} 0 \\ \mu \end{pmatrix}$, so that

$$F(x, u) = \begin{pmatrix} \alpha_2 \\ -3.09 \sin \alpha_1 + \mu \end{pmatrix}. \quad (3.3.2)$$

By (3.1.3), the cost function is

$$G(x, u) = \alpha_1^2 + \alpha_2^2 + \mu^2, \quad (3.3.3)$$

so that

$$V(X(t)) = -\int_0^{t_f} (\alpha_1^2(s) + \alpha_2^2(s) + \mu^2(s)) ds, \quad (3.3.4)$$

and by $\frac{\partial V}{\partial x} \frac{\partial F(x, u)}{\partial u} + \frac{\partial G(x, u)}{\partial u} = 0$, we have

$$\begin{pmatrix} \frac{\partial V}{\partial \alpha_1} & \frac{\partial V}{\partial \alpha_2} \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} + 2\mu = 0, \quad (3.3.5)$$

and hence the control can be computed as

$$\mu = -\frac{1}{2} \left(\frac{\partial V}{\partial \alpha_2} \right), \quad (3.3.6)$$

or rather

$$u = \begin{pmatrix} 0 \\ \mu \end{pmatrix} = \begin{pmatrix} 0 & -\frac{1}{2} \left(\frac{\partial V}{\partial \alpha_2} \right) \end{pmatrix}^T. \quad (3.3.7)$$

3.3.2 Application of Kleinman-Newton Method

We choose a stabilizing control by guessing

$$u_0(x) = \alpha_1 - 2\alpha_2 = (1 \quad -2) \begin{pmatrix} \alpha_1 \\ \alpha_2 \end{pmatrix} \quad (3.3.8)$$

as the initial control, and chose the subspace

$$D = \{(x_1, x_2) : x_1 \in [-1, 1], x_2 \in [-2, 2]\}. \quad (3.3.9)$$

The grid length $h=0.05$, and the starting and final time are $t_0=0$ and $t_f=15$ respectively. Therefore there will be $M = \left(\frac{2-(-2)}{0.05} + 1\right) \times \left(\frac{1-(-1)}{0.05} + 1\right) = 3321$ initial conditions or rather grid points which are the vertices of finite elements. They are listed in Figure 3.1 and denoted by $\{x_0^1, x_0^2, \dots, x_0^M\}$.

$$\begin{array}{cccc} (-1, -2), & (-0.95, -2), & \dots, & (1, -2) \\ (-1, -1.95), & (-0.95, -1.95), & \dots, & (1, -1.95) \\ \vdots & \vdots & & \vdots \\ (-1, 2), & (-0.95, 2), & \dots, & (1, 2) \end{array}$$

Figure 3.3.1: All the grid points in the chosen domain D .

With these M initial conditions and the initial controls, we use second order Runge Kutta method to solve (3.3.1) for M trajectories. The iteration of Runge Kutta method will be nestled inside the iteration of Kleinman-Newton method.

The iteration of Runge Kutta method is described in (3.3.10). Let $\begin{pmatrix} \alpha_1(n) \\ \alpha_2(n) \end{pmatrix}$ be the discretized solution (trajectory) with $n = 0, 1, 2, \dots, N$, so that $\begin{pmatrix} \alpha_1(0) \\ \alpha_2(0) \end{pmatrix}$ is the initial value following the pattern in Figure 3.1. Since the starting time is 0 and final time is 15, if we let the step size be 0.05 the same as the grid length, then the number of discretized solution is $N = \frac{15-0}{0.05} + 1 = 301$.

$$\left\{ \begin{array}{l} S = \begin{pmatrix} \alpha_2 + \frac{h}{2} [\mu(n) - a \sin \alpha_1(n)] \\ -a \sin \left(\alpha_1(n) + \frac{h\alpha_2(n)}{2} \right) + \mu(n) \end{pmatrix}; \\ \begin{pmatrix} \alpha_1(n+1) \\ \alpha_2(n+1) \end{pmatrix} = \begin{pmatrix} \alpha_1(n) \\ \alpha_2(n) \end{pmatrix} + hS. \end{array} \right. \quad (3.3.10)$$

where the iteration number $n = 0, 1, 2, 3, \dots, 300$, step size $h = 0.05$, and S is a parameter.

Note: We have $M = 3321$ such trajectories because of the pattern in Figure 3.1, which means we will have to solve equation (3.3.10) 3321 times, and during each time there are $N = 301$ iterations of Runge Kutta method nested.

3.3.3 Iteration of Kleinman-Newton Method

- 1) Start with 3321 initial values in Figure 3.1, by iterating (3.3.10) to get 3321 trajectories, using initial control (3.3.8) in the first iteration and using the generated control values by (3.3.7) from the second iteration on.

- 2) Integrate (3.3.4) along those 3321 trajectories by *Simpson's rule* or the *Gaussian quadrature method* to get 3321 values of V (see Figure 3.2).
- 3) Use finite difference method (3.3.11) to differentiate V with respect to α_2 to get 3321 values of control μ to form $u = \begin{pmatrix} 0 \\ \mu \end{pmatrix}$ by (3.3.7).

$$\mu^i = -\frac{1}{2} \left(\frac{\partial V_k^i}{\partial \alpha_2} \right) = -\frac{1}{4h} (V_k^{i-1} - V_k^{i+1}), \quad (3.3.11)$$

where k is the number of iteration of Kleinman-Newton method, and $i \leq 3321$ is the location on the pattern in Figure 3.2.

Note: By (3.3.11) we see that we can not calculate the boundary values of μ^i , so that we just simply let the boundary values of μ^i equal to the control values right next to them.

- 4) Go to 1) to start next iteration with initial values in Figure 3.1 and with the newly generated control values by (3.3.7).
- 5) Stop when the iteration number accumulates to the specified threshold (usually 4 is large enough, but we implemented 8 iterations for the sake of research).

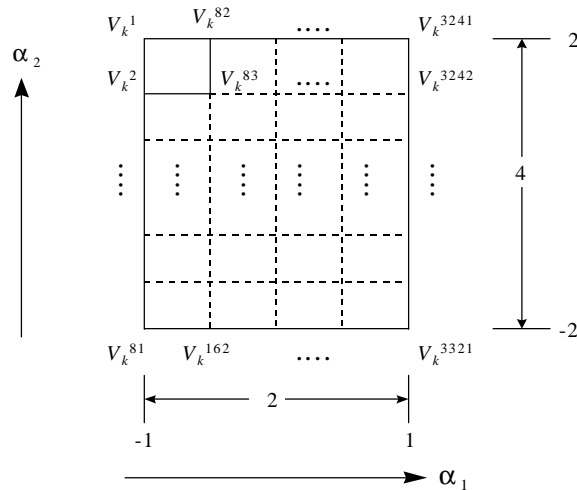


Figure 3.3.2: The pattern of the values of V at iteration step k .

3.3.4 Interpolation of Control Values

When solving (3.3.10) by Runge Kutta method, every solution value $x(n) = \begin{pmatrix} \alpha_1(n) \\ \alpha_2(n) \end{pmatrix}$ needs a corresponding control value $u(n) = \begin{pmatrix} 0 \\ \mu(n) \end{pmatrix}$. If a solution value $x(n)$ is not at grid point, there will be 4 values of the control u surrounding $x(n)$ (see Figure 3.3).

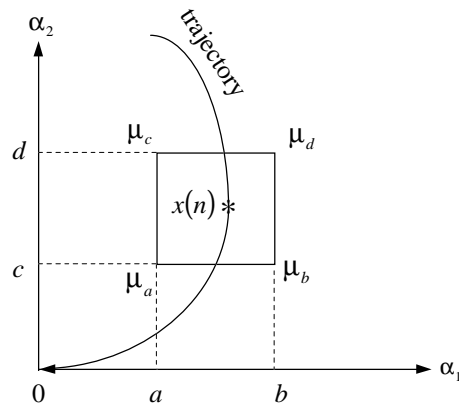


Figure 3.3.3: There are 4 control values surrounding a solution value $x(n)$.

In order to keep high accuracy of computation, none of the 4 control values will be used directly, but instead the corresponding control value $\mu(n)$ will be found by linear interpolation according equation (3.3.12).

$$\begin{aligned}
\mu(n) = & \mu_a \left(1 + \frac{a}{b-a} - \frac{\alpha_1}{b-a} \right) \left(1 + \frac{c}{d-c} - \frac{\alpha_2}{d-c} \right) \\
& + \mu_b \left(1 + \frac{c}{d-c} - \frac{\alpha_2}{d-c} \right) \left(\frac{\alpha_1}{b-a} - \frac{a}{b-a} \right) \\
& + \mu_c \left(1 + \frac{a}{b-a} - \frac{\alpha_1}{b-a} \right) \left(\frac{\alpha_2}{d-c} - \frac{c}{d-c} \right) \\
& + \mu_d \left(\frac{\alpha_1}{b-a} - \frac{a}{b-a} \right) \left(\frac{\alpha_2}{d-c} - \frac{c}{d-c} \right).
\end{aligned} \tag{3.3.12}$$

3.3.5 A Brief Description of Data Structure

There are several technical problems need to be solved:

- 1) How to keep the values of x , V , and u .
- 2) How to locate where the trajectory goes in α_1 - α_2 phase plane in order to know which 4 control values are needed for the interpolation.
- 3) How to prevent the trajectory from going beyond the domain D in α_1 - α_2 phase plane.
- 4) Since the finite difference method (3.3.11) which is used to take derivative with respect to α_2 does not deal with control values on the boundary, what should we do with these boundary control values.

The first problem is solved by using “linked mesh” technique to maintain the whole data structure (see Figure 3.4). Each node contains information of trajectory value x and control value u , but not integrated value of V because it is not used during computations.

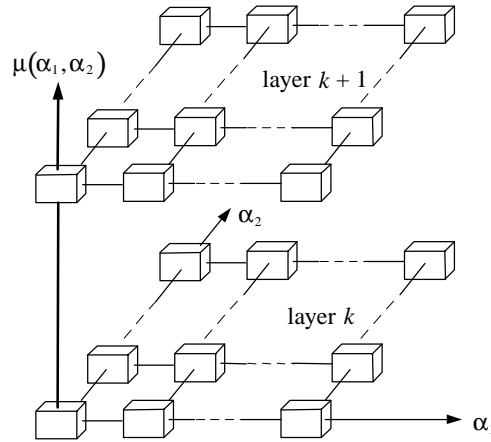


Figure 3.3.4: Two layers of the whole data structure: the node is linked with one another within the same layer, and each layer is linked with one another by the corner node.

With this “linked mesh” technique, as we iterate (3.3.10), we first make the pointer search along $\mu(\alpha_1, \alpha_2)$ for which layer the trajectory has reached according to the control value $\mu(n)$, and then let the pointer search within the same layer according to the trajectory value $x(n)$ for which 4 nodes are needed for interpolation. Now problem 2 is solved.

Since outside the domain D , there will be no control values stored in the data structure nodes, hence at this time (3.3.10) can not be solved. In order to prevent the trajectory from going beyond the domain D in α_1 - α_2 phase plane, we shrink the boundary of domain D by one or two grid size after each Kleinman-Newton iteration. After this shrinking, if the trajectory still goes beyond domain D , it will not go too far, so that we will use 4 nodes that are closest to the current trajectory location $x(n)$ for the interpolation of the corresponding control value $\mu(n)$. The resulting error is still tolerable.

As for problem 4, we just simply let each boundary control value equal to the one stored in the next inner node within the same layer.

3.3.6 Computation Result

After about 8 iterations, all the controls will converge to $\{\hat{u}^1, \hat{u}^2, \dots, \hat{u}^M\}$ which have the optimal cost. Figure 3.6 (a), (b), (c), (d), (e), (f), (g), and (h) are the output of $V(x)$ from iteration 0 to 7 respectively. Figure 3.7 (a), (b), (c), (d), (e), (f), (g), and (h) are the output of $\mu(x)$ from iteration 0 to 7 respectively. We can clearly see the convergence of $V(x)$ and $\mu(x)$.

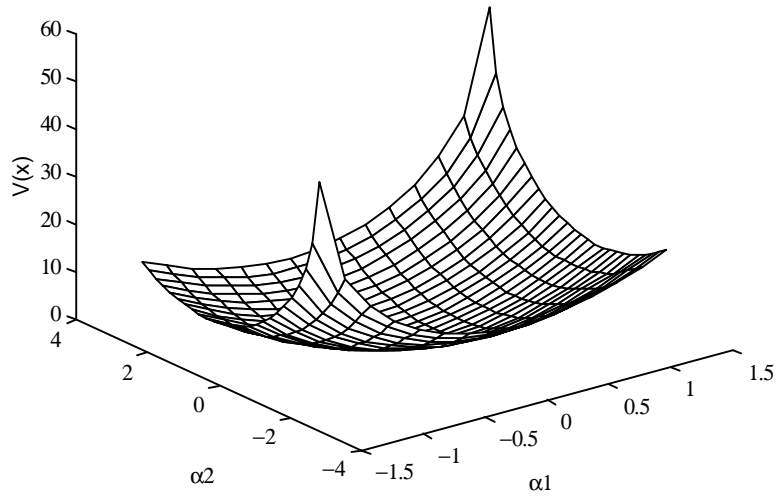


Figure 3.3.5: Output of $V(x)$ in iteration 0.

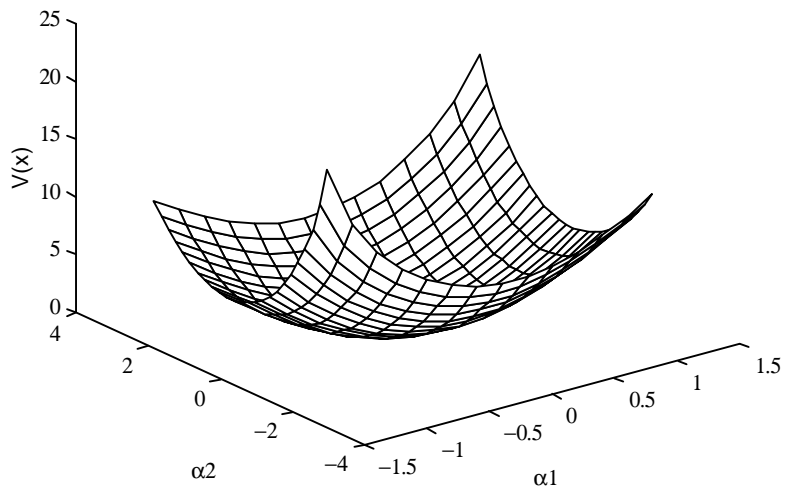


Figure 3.3.6: Output of $V(x)$ in iteration 1.

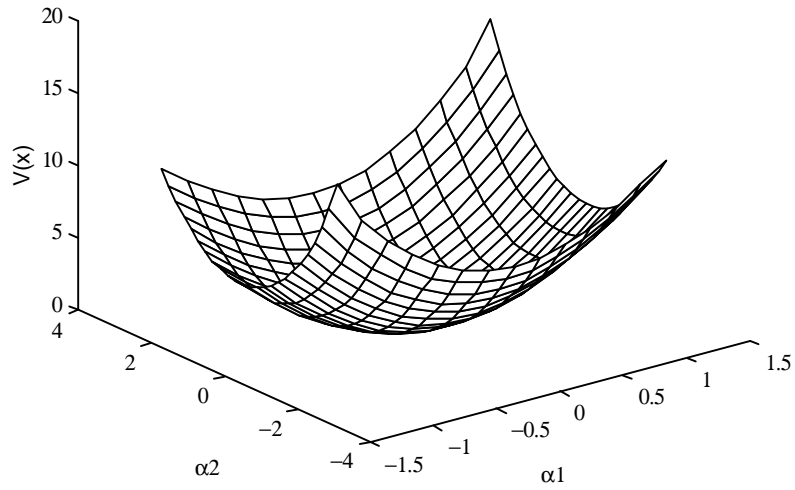


Figure 3.3.7: Output of $V(x)$ in iteration 2.

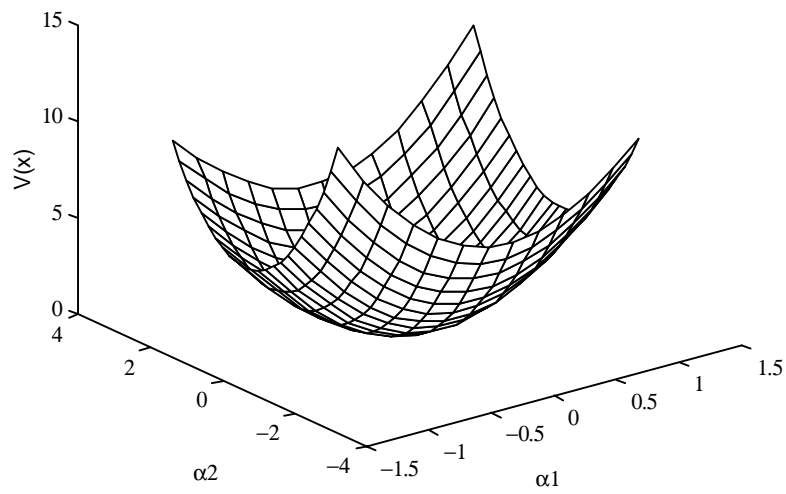


Figure 3.3.8: Output of $V(x)$ in iteration 3.

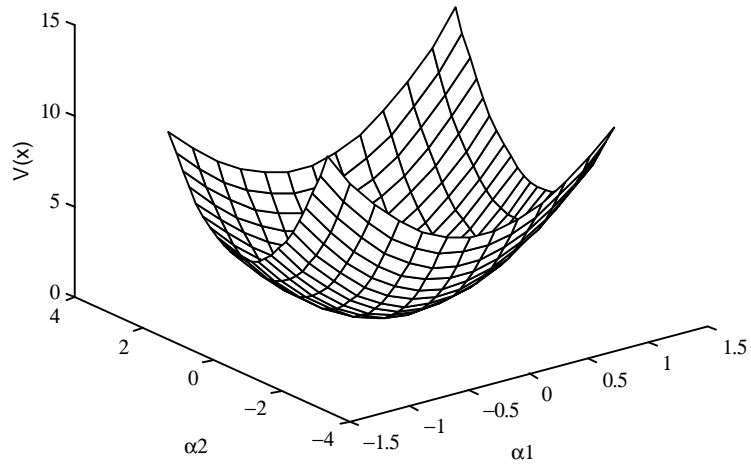


Figure 3.3.9: Output of $V(x)$ in iteration 4.

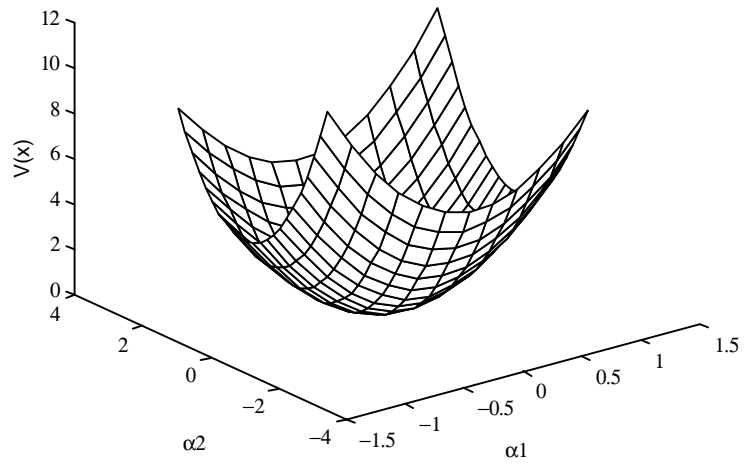


Figure 3.3.10: Output of $V(x)$ in iteration 5.

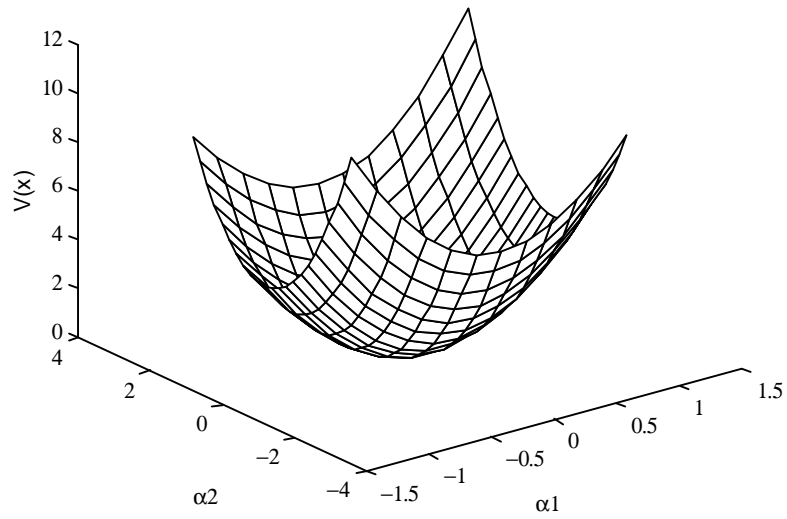


Figure 3.3.11: Output of $V(x)$ in iteration 6.

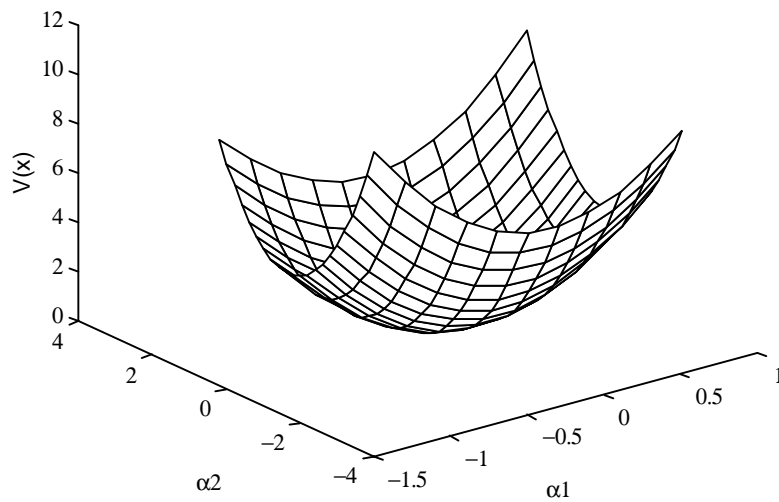


Figure 3.3.12: Output of $V(x)$ in iteration 7.

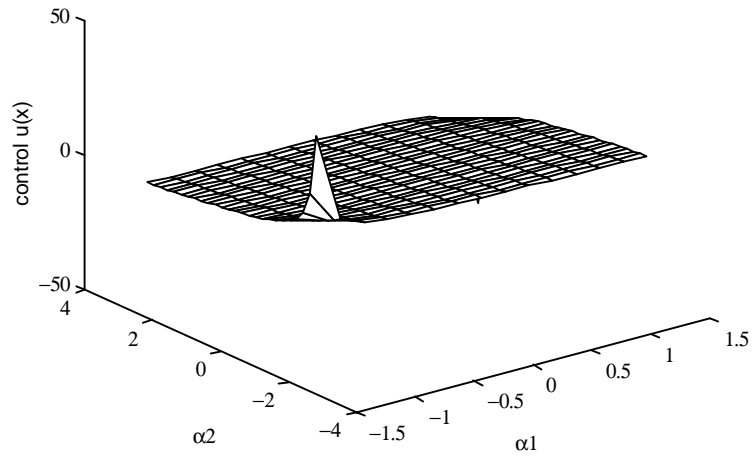


Figure 3.3.13: Output of $\mu(x)$ in iteration 0.

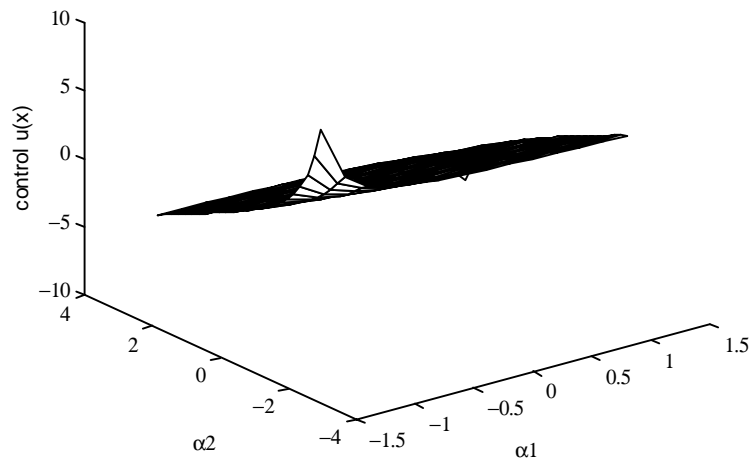


Figure 3.3.14: Output of $\mu(x)$ in iteration 1.

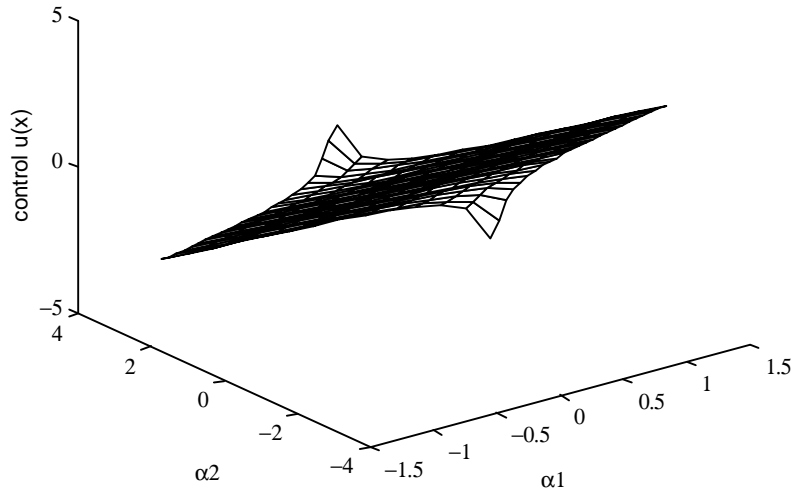


Figure 3.3.15: Output of $\mu(x)$ in iteration 2.

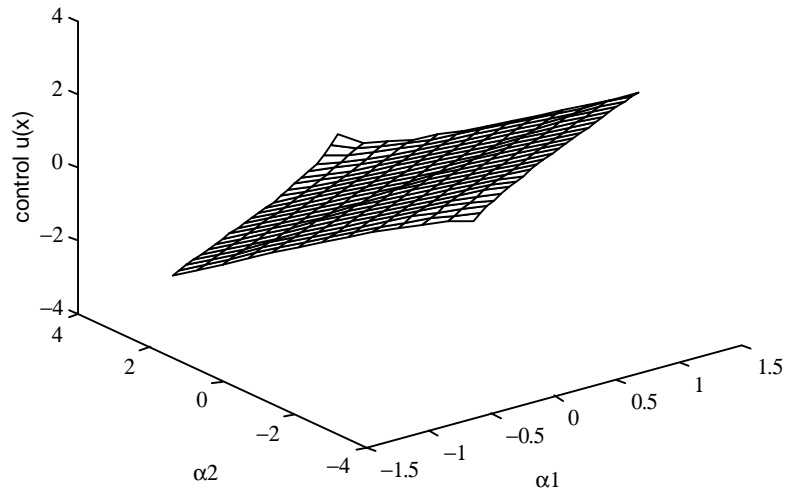


Figure 3.3.16: Output of $\mu(x)$ in iteration 3.

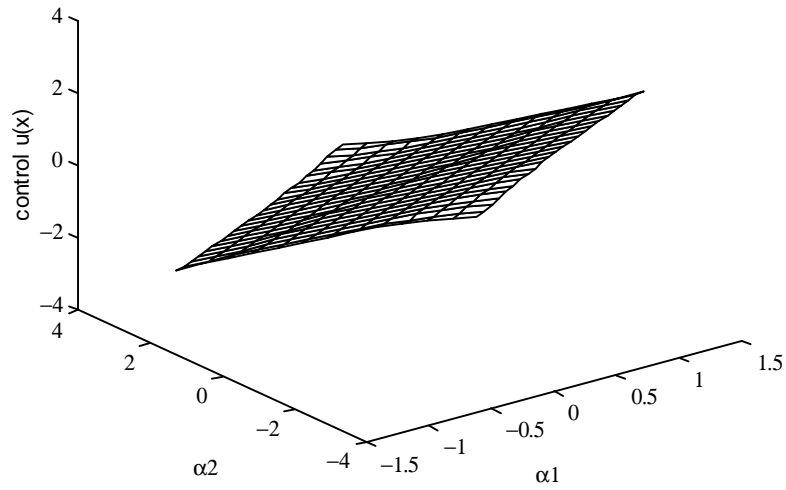


Figure 3.3.17: Output of $\mu(x)$ in iteration 4.

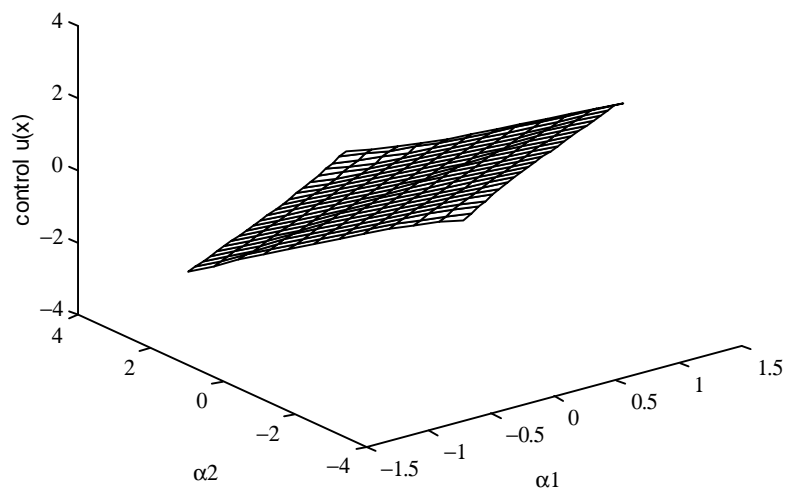


Figure 3.3.18: Output of $\mu(x)$ in iteration 5.

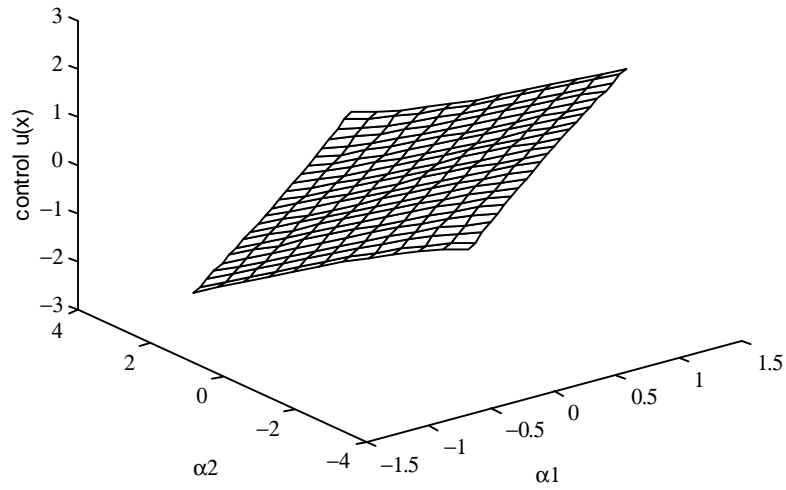


Figure 3.3.19: Output of $\mu(x)$ in iteration 6.

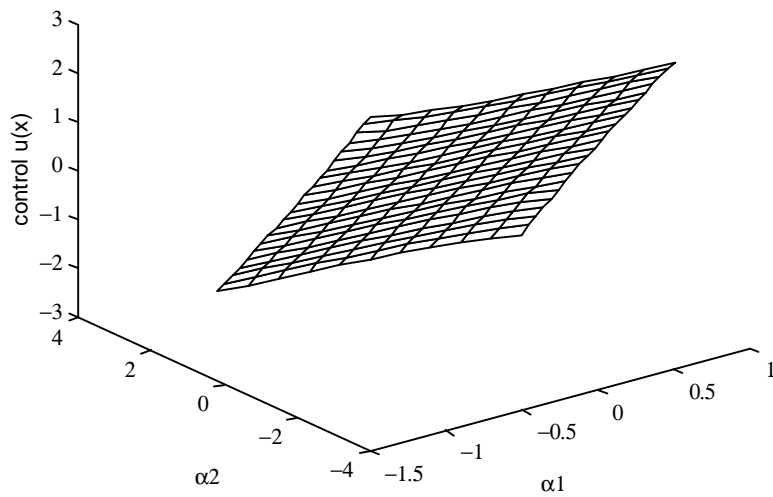


Figure 3.3.20: Output of $\mu(x)$ in iteration 7.

In order to study a certain individual trajectory, we have recorded the results of four different initial conditions which are in 4 different quadrants: $(-0.5, -1)$, $(0.5, 1.3)$, $(-0.5, 1.2)$, $(0.5, -1.1)$. Figure 3.8 (a) and (b) are the α_1 - α_2 phase plane which show the situation of convergence of the four trajectories starting from those four initial values. Figure 3.9 (a), (b), (c), and (d) show the corresponding control $\mu(x(t))$ with respect time t , and under those 4 different situations, the controls in each iteration converge to the optimal one.

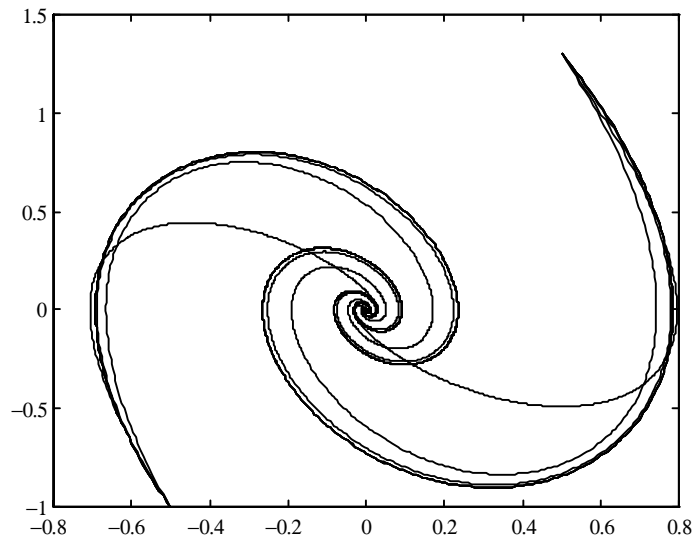


Figure 3.3.21: The phase plane with initial values of $(-0.5, -1)$ and $(0.5, 1.3)$.

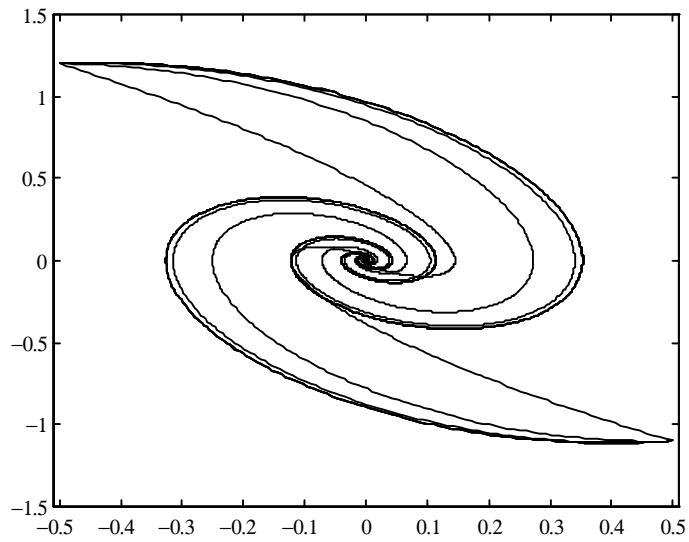


Figure 3.3.22: The phase plane with initial values of $(-0.5, 1.2)$ and $(0.5, -1.1)$.

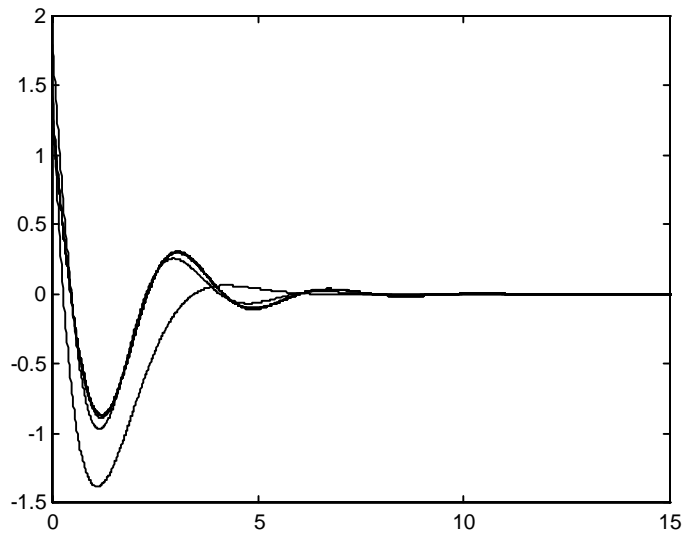


Figure 3.3.23: Control $\mu(x(t))$ corresponds to the initial value of $(-0.5, -1)$.

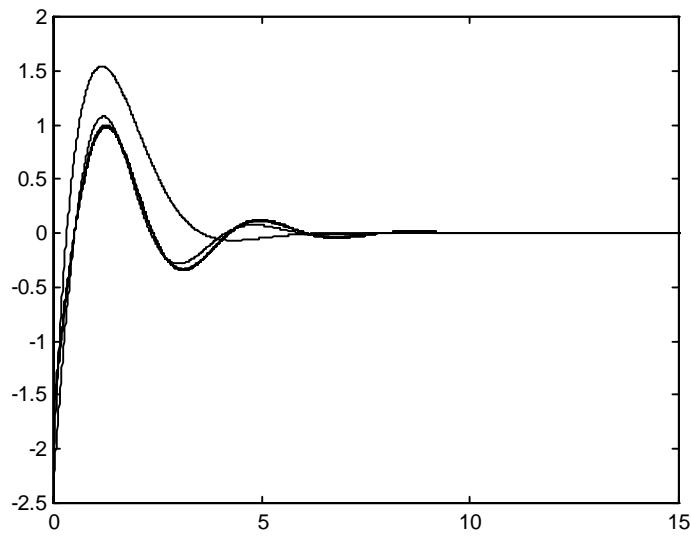


Figure 3.3.24: Control $\mu(x(t))$ corresponds to the initial value of $(0.5, 1.3)$.

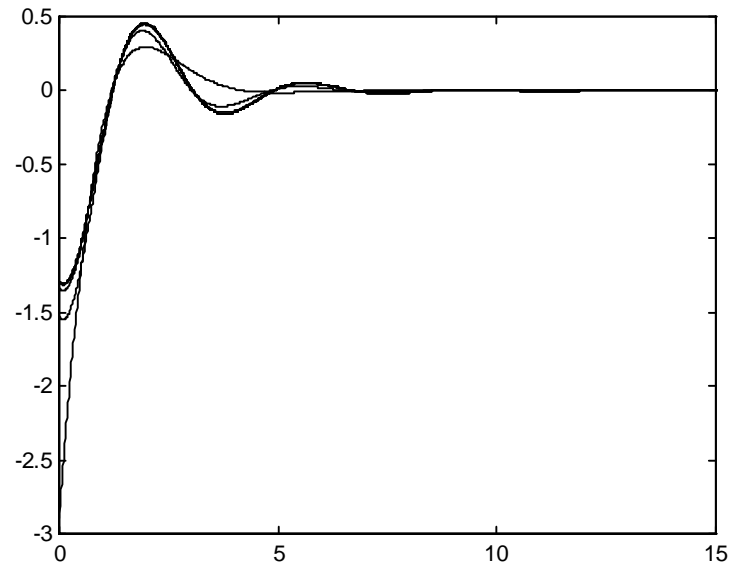


Figure 3.3.25: Control $\mu(x(t))$ corresponds to the initial value of $(-0.5, 1.2)$.

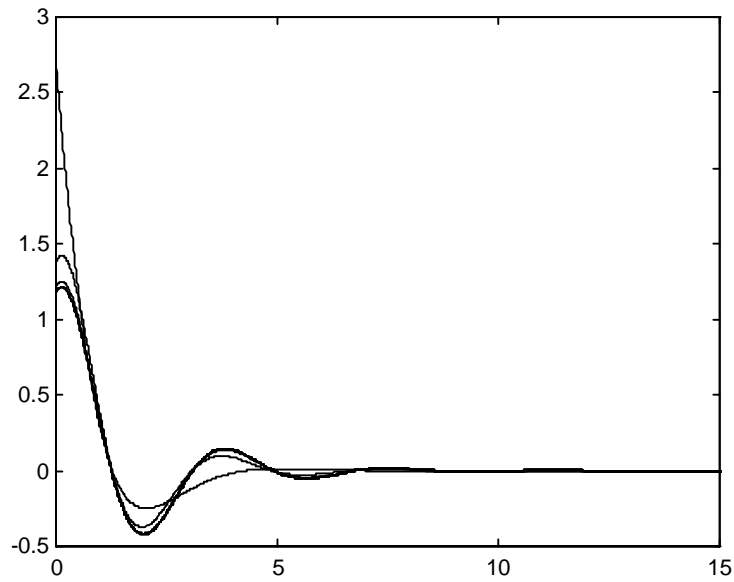


Figure 3.3.26: Control $\mu(x(t))$ corresponds to the initial value of $(0.5, -1.1)$.

Further more, from table 3.1, we can see the cost drops to the minimum one and become stable. table 3.2 shows the comparison between the theoretical cost and the numerical cost. The theoretical cost is computed by the Ricotta equation described in Chapter 1 section 1.3, and the numerical cost is computed by Simpson integration rule following the equation (3.3.4).

Table 3.3.1: The cost of the four trajectories calculated according to the definition (3.3.4).

| Loop No. \ $x(0)$ | $(-0.5, -1)$ | $(0.5, 1.3)$ | $(-0.5, 1.2)$ | $(0.5, -1.1)$ |
|-------------------|--------------|--------------|---------------|---------------|
| 0 | 3.5386 | 4.7625 | 2.0400 | 2.7498 |
| 1 | 2.3272 | 3.2277 | 2.4364 | 2.1805 |
| 2 | 2.2357 | 3.0873 | 2.3705 | 2.1234 |
| 3 | 2.2347 | 3.0857 | 2.3694 | 2.1224 |
| 4 | 2.2347 | 3.0857 | 2.3694 | 2.1224 |
| 5 | 2.2347 | 3.0857 | 2.3694 | 2.1224 |
| 6 | 2.2347 | 3.0857 | 2.3694 | 2.1224 |
| 7 | 2.2347 | 3.0857 | 2.3694 | 2.1224 |

The theoretical costs computed by Ricatti equation and the numerical costs are as followings:

Table 3.3.2: The comparison between the four resulted theoretical cost and numerical cost.

| $x(0) \setminus \text{Cost}$ | Numerical | Theoretical | Absolute Error | Relative Error |
|------------------------------|-----------|-------------|----------------|----------------|
| $(-0.5, -1)$ | 2.234736 | 2.236646 | 0.00191 | 0.085% |
| $(0.5, 1.3)$ | 3.085725 | 3.074801 | 0.0109 | 0.335% |
| $(-0.5, 1.2)$ | 2.369434 | 2.393589 | 0.02461 | 1.009% |
| $(0.5, -1.1)$ | 2.122434 | 2.145561 | 0.02313 | 1.078% |

The absolute error is defined as the absolute value of the difference between theoretical cost and numerical cost, and the relative error is defined as $\frac{\text{Relative Error}}{\text{Theoretical Cost}}$.

3.4 Conclusion

The Kleinman-Newton method has come out of linear quadratic case to the stage of nonlinear-nonquadratic situation. It can be applied to any dimensional space. The convergence rate seems to be quadratic, but it needs more research work. Whether the domain D can be extended to the global domain also needs further exploration. The numerical results match very well the corresponding theoretical results.

Reference

- [1] B. D. O. Anderson and J. B. Moor, "Linear Optimal Control," Prentice Hall Inc., Englewood Cliff, N. J., 1971.
- [2] Cronin, Jane, "Differential Equations," Marcel Dekker, Inc., 1994.
- [3] Lee, B. E. and Markus, L., "Foundations of Optimal Control Theory," John Wiley & Sons, Inc. 1967.
- [4] Kleinman, D. L., "On an Iterative Technique for Riccati Equation Computations," IEEE Transactions on Automatic Control, February 1968.
- [5] Lukes, D. L., "Optimal Regulation of Nonlinear Dynamical Systems," SIAM J. Control, Vol. 7, No. 1, February 1969.
- [6] Pontryagin, V. G. Boltyanskii, R. V. Gamkrelidze, and E. F. Mishchenko, "The Mathematical Theory of Optimal Process," John Wiley & Sons, Inc. 1962.
- [7] J. E. Potters, "Matrix quadratic solutions," J. SIAM, Mathematics, April 14, 1966.
- [8] Russell, D. L., "Mathematics of Finite-Dimensional Control Systems," Marcel Dekker, Inc., 1979.
- [9] Zhang, Xiaohong, "Optimal Feedback Control for Nonlinear Discrete Systems and Applications to Optimal Control of Nonlinear Periodic Ordinary Differential Systems," Ph.D. Dissertation in Mathematics, Department of Mathematics, Virginia Tech, February 1993.
- [10] Zhu, Jinghao, "Some Results on Nonlinear Optimal Control", Ph.D. Dissertation in Mathematics, Department of Mathematics, Virginia Tech, 1996.

Vita

Jinghong Kang was born on December 29, 1964 in Beijing, China. He graduated from Beijing University of Science and Technology in 1987 with a Bachelor of Engineering in Metal Physics. He entered Eastern Illinois University in 1991, and got his Master of Arts in Mathematics in May 1992. He entered Virginia Polytechnic Institute and State University in August 1992, and since then he has been working with Dr. David Russell towards his Ph.D. degree in Mathematics.