SCHOOL OF OPERATIONS RESEARCH
AND INDUSTRIAL ENGINEERING
COLLEGE OF ENGINEERING
CORNELL UNIVERSITY
ITHACA, NY 14853

TECHNICAL REPORT NO. 917

August 1990

THE PROOF OF QUADRATIC CONVERGENCE OF DIFFERENTIAL DYNAMIC PROGRAMMING

by

Li-zhi Liao
and
Christine A. Shoemaker

Li-zhi Liao: Graduate Research Assistant, School of Operations Research and Industrial Engineering, Upson Hall, Cornell University, Ithaca, N.Y. 14853, (607)-255-9139.

Christine A. Shoemaker: Professor, School of Civil and Environmental Engineering, Graduate Fields of Operations Research and of Applied Mathematics Center, Hollister Hall, Cornell University, Ithaca, N.Y. 14853, (607)-255-9233.

This work was supported in part by N.S.F. (Directorate on Computer and Informational Sciences) ASC 8915326 and by the U.S.A.F., University Research Initiative on Nonlinear Dynamics and Control of Flexible Structures at Cornell University.

The Proof of Quadratic Convergence of Differential Dynamic Programming

Li-zhi Liao and Christine A. Shoemaker

Abstract. In this report, we will provide another proof for the quadratic convergence of Differential Dynamic Programming. This proof is based on the dynamic programming and Bellman Optimality Principle, and is derived directly from the DDP algorithm steps.

Introduction

The unconstrained discrete-time optimal control problem studied in this report has the following format:

$$\min_{(u_1,\cdots,u_N)} \sum_{t=1}^{N} g(x_t, u_t, t) \tag{P_0}$$

$$\text{where} \quad x_{t+1} = f(x_t, u_t, t) \quad t = 1, \cdots, N,$$
$$x_1 \equiv \bar{x}_1 \quad \text{given and fixed}$$

where $x_t \in R^n$ and $u_t \in R^m$ are called state and control variables; the function $g : R^{n+m+1} \rightarrow R^1$ is called the objective function (or performance index); and the function $f : R^{n+m+1} \rightarrow R^n$ is called the transition function.

The first algorithm that solves general problem (P_0) and captures its special structures was proposed by Mayne [4] in 1966. His algorithm, which is called Differential Dynamic Programming (**DDP**), combined the Dynamic Programming (**DP**) scheme and Newton's method. His original DDP algorithm was further developed later by Jacobson and Mayne [1]. Yakowitz [7] gave a good survey on the development of the DDP algorithm.

This technical report is a companion to a paper by Liao and Shoemaker [3] to be published in *IEEE Trans. on Automatic Control.* An abbreviated version of the proof from this technical report appears in [3]. It is the purpose of this document to provide a more detailed description of the proof of quadratic convergence of DDP than is available in [3].

Quadratic Convergence Proof of the DDP Algorithm

Since DDP method was proposed by Mayne in 1966, there was no rigorous convergence proof until the early 1980's. The global convergence proof for the DDP algorithm was given by Yakowitz and Rutherford [8]. Proofs of quadratic convergence of the DDP algorithm were given independently by Pantoja [6] in 1983, and Murray and Yakowitz [5] in 1984. Both proofs were based on the comparison between DDP method and Newton's method. Neither of the two proofs used dynamic programming and Bellman's Optimality Principle, which are the basis of the DDP method. Here, we will give a proof of the quadratic convergence of the DDP algorithm. This proof follows the same track as the development of the DDP algorithm and is more straightforward. The result of the quadratic convergence of the DDP algorithm is summarized in the following Theorem. But, first we need the following notation. Define

$$\vec{U} \doteq (u_1^T, \cdots, u_N^T)^T \in R^{Nm}; \quad \vec{X} \doteq (x_1^T, \cdots, x_N^T)^T \in R^{Nn}$$

and call $\vec{U}$ the policy and $\vec{X}$ the trajectory associated with policy $\vec{U}$.

Theorem: Assumptions:

1) $g(x_t, u_t, t)$ $t = 1, \ldots, N$ and $f(x_t, u_t, t)$ $t = 1, \cdots, N-1$ have continuous third partial derivatives with respect to x_t and u_t over a closed bounded convex set $D \subset R^{n+m}$.

2) $\{\vec{U}^{(l)}\}$ is the policy sequence obtained by the lth iteration of the DDP algorithm.

3) The matrices $C_t(\vec{U}^{(l)})$, $t = 1, \cdots, N$, $l \geq 0$ computed in the lth iteration of the DDP algorithm are all positive definite in D.

4) $(x_t^{(l)}, u_t^{(l)}) \in D$ for all $l > 0$ and for $t = 1, \cdots, N$ where $x_t^{(l)}$ and $u_t^{(l)}$ are components of $\vec{X}^{(l)}$ and $\vec{U}^{(l)}$ computed in the lth iteration of DDP.

5) $\{\vec{U}^{(l)}\}$ converges to $\vec{U}^*$ which is a solution to problem (P_0), where $\vec{X}^*$ is the trajectory associated with $\vec{U}^*$ and $(x_t^*, u_t^*) \in int(D)$ for $t = 1, \cdots, N$.

Let $\vec{U}^{(0)}$ be the initial policy, $\vec{X}^{(0)}$ be the trajectory associated with $\vec{U}^{(0)}$, and $(x_t^{(0)}, u_t^{(0)}) \in D$ for $t = 1, \cdots, N$.

Conclusion: There exists a constant $c > 0$ such that

$$\|\vec{U}^{(l+1)} - \vec{U}^*\| \leq c \cdot \|\vec{U}^{(l)} - \vec{U}^*\|^2 \qquad \text{for all } l \geq 0 \tag{1}$$

provided $\|\vec{U}^{(0)} - \vec{U}^*\|$ is sufficiently small.

Hence the DDP method converges quadratically in D.

For convenience, we will give the proof of this theorem in the case of $m = n = 1$. The multi-dimensional proof follows the same steps as the scalar proof described here. A similar theorem is given earlier by Murray and Yakowitz in [5], but the proof given here is different and is derived directly from the DDP algorithm steps in the Appendix, whereas the earlier proof in [5] is done by comparison to the Newton's solution to problem (P_0).

Proof: For any $l \geq 0$, let vectors $\bar{\vec{U}}(k) = \vec{U}^{(l)}$, $\vec{U}(k) = \vec{U}^{(l+1)}$ be the solutions resulting from the lth and $(l+1)$th iterations of the DDP procedure, respectively, for solving (2.1) with $N = k$. Also, let $\vec{U}^*(k)$ denote the optimal solution to problem (2.1) with $N = k$, and define

$$\begin{cases} g^t(x,u) \doteq g(x,u,t) & t = 1, \cdots, N \\ f^t(x,u) \doteq f(x,u,t) & t = 1, \cdots, N-1 \end{cases} \tag{2}$$

and the power index will be put as the superscript outside parentheses. We will use induction on the number of time steps N to prove this theorem. Our induction hypothesis is that given 1) to 5) above, then (1) is true for a given value of N.

a) First, for $N = 2$, consider

$$\min_{u_1, u_2} \; J(x_1, u_1, x_2, u_2) = \; [g^1(x_1, u_1) + g^2(x_2, u_2)] \tag{3}$$

where $x_2 = f^1(x_1, u_1)$,

$x_1 \equiv \bar{x}_1$, $(\bar{u}_1, \bar{u}_2)$ initial policy.

The application of the DDP algorithm to problem (3) results in

$$A_2 = \frac{1}{2} g^2_{xx}, \quad B_2 = g^2_{xu}, \quad C_2 = \frac{1}{2} g^2_{uu}, \quad D_2 = g^2_u, \quad E_2 = g^2_x, \tag{4}$$

(From assumption 3) and equation (4), we know $\quad C_2 > 0 \quad$ or $\quad g^2_{uu} > 0.$)

$$P_2 = \frac{1}{2} g^2_{xx} - \frac{1}{2} \frac{(g^2_{xu})^2}{g^2_{uu}}, \qquad Q_2 = -\frac{g^2_u \cdot g^2_{xu}}{g^2_{uu}} + g^2_x, \tag{5}$$

$$\alpha_2 = -\frac{g^2_u}{g^2_{uu}}, \qquad \beta_2 = -\frac{g^2_{xu}}{g^2_{uu}}, \tag{6}$$

$$A_1 = \frac{1}{2} \{ g^1_{xx} + 2(f^1_x)^2 \cdot [\frac{1}{2} g^2_{xx} - \frac{1}{2} \frac{(g^2_{xu})^2}{g^2_{uu}}] + [-\frac{g^2_u \cdot g^2_{xu}}{g^2_{uu}} + g^2_x] \cdot f^1_{xx} \}, \tag{7}$$

$$B_1 = g^1_{xu} + 2 f^1_u \cdot f^1_x \cdot [\frac{1}{2} g^2_{xx} - \frac{1}{2} \frac{(g^2_{xu})^2}{g^2_{uu}}] + [-\frac{g^2_u \cdot g^2_{xu}}{g^2_{uu}} + g^2_x] \cdot f^1_{xu}, \tag{8}$$

$$C_1 = \frac{1}{2}\{g^1_{uu} + 2(f^1_u)^2 \cdot [\frac{1}{2}g^2_{xx} - \frac{1}{2}\frac{(g^2_{xu})^2}{g^2_{uu}}] + [-\frac{g^2_u \cdot g^2_{xu}}{g^2_{uu}} + g^2_x] \cdot f^1_{uu}\}, \tag{9}$$

$$D_1 = g^1_u + f^1_u \cdot [-\frac{g^2_u \cdot g^2_{xu}}{g^2_{uu}} + g^2_x], \tag{10}$$

$$E_1 = g^1_x + f^1_x \cdot [-\frac{g^2_u \cdot g^2_{xu}}{g^2_{uu}} + g^2_x], \tag{11}$$

(From assumption 3) and equation (9), we know $C_1 > 0.$)

$$\alpha_1 = -\frac{1}{2C_1}D_1, \qquad \beta_1 = -\frac{1}{2C_1}B_1. \tag{12}$$

Since $(\bar{x}_1, u_1^*, x_2^*, u_2^*)$ is a solution to problem (3), then

$$J(\bar{x}_1, u_1^*, x_2^*, u_2^*) = \min_{u_1,u_2} J(\bar{x}_1, u_1, x_2, u_2). \tag{13}$$

So,
$$J_{u_2}(\bar{x}_1, u_1^*, x_2^*, u_2^*) = g^2_u|_{(x_2^*,u_2^*)} = 0, \tag{14}$$
$$J_{u_1}(\bar{x}_1, u_1^*, x_2^*, u_2^*) = (g^1_u + g^2_x \cdot f^1_u)|_{(\bar{x}_1,x_2^*,u_1^*,u_2^*)} = 0. \tag{15}$$

From Taylor expansion, we have

$$g^2_u|_{(\bar{x}_2,\bar{u}_2)} - g^2_u|_{(x_2^*,u_2^*)}$$
$$= g^2_{uu} \cdot (\bar{u}_2 - u_2^*) + g^2_{xu} \cdot (\bar{x}_2 - x_2^*) + O\big([(\bar{x}_2 - x_2^*) + (\bar{u}_2 - u_2^*)]^2\big) \tag{16}$$

where $\bar{x}_2 = f^1(\bar{x}_1, \bar{u}_1)$.

Since
$$\bar{x}_2 - x_2^* = f^1_u \cdot (\bar{u}_1 - u_1^*) + O\big((\bar{u}_1 - u_1^*)^2\big), \tag{17}$$

then
$$g^2_u|_{(\bar{x}_2,\bar{u}_2)} - g^2_u|_{(x_2^*,u_2^*)} = g^2_{uu} \cdot (\bar{u}_2 - u_2^*) + g^2_{xu} \cdot f^1_u \cdot (\bar{u}_1 - u_1^*)$$
$$+O\big([(\bar{u}_1 - u_1^*) + (\bar{u}_2 - u_2^*)]^2\big). \tag{18}$$

Similarly, we have

$$(g^1_u + g^2_x \cdot f^1_u)|_{(\bar{x}_1,\bar{x}_2,\bar{u}_1,\bar{u}_2)} - (g^1_u + g^2_x \cdot f^1_u)|_{(\bar{x}_1,x_2^*,u_1^*,u_2^*)}$$
$$= g^1_{uu} \cdot (\bar{u}_1 - u_1^*) + g^2_{xx} \cdot (f^1_u)^2 \cdot (\bar{u}_1 - u_1^*) + g^2_{xu} \cdot f^1_u \cdot (\bar{u}_2 - u_2^*)$$
$$+g^2_x \cdot f^1_{uu} \cdot (\bar{u}_1 - u_1^*) + O\big([(\bar{u}_1 - u_1^*) + (\bar{u}_2 - u_2^*)]^2\big). \tag{19}$$

By equations (14) and (18), we have

$$g_u^2|_{(\bar{x}_2,\bar{u}_2)} = g_{uu}^2 \cdot (\bar{u}_2 - u_2^*) + g_{xu}^2 \cdot f_u^1 \cdot (\bar{u}_1 - u_1^*) + O\big([(\bar{u}_1 - u_1^*) + (\bar{u}_2 - u_2^*)]^2\big). \quad (20)$$

By equations (15) and (19), we have

$$(g_u^1 + g_x^2 \cdot f_u^1)|_{(\bar{x}_1,\bar{x}_2,\bar{u}_1,\bar{u}_2)} = (g_{uu}^1 + g_{xx}^2 \cdot (f_u^1)^2 + g_x^2 \cdot f_{uu}^1) \cdot (\bar{u}_1 - u_1^*)$$

$$+ g_{xu}^2 \cdot f_u^1 \cdot (\bar{u}_2 - u_2^*) + O\big([(\bar{u}_1 - u_1^*) + (\bar{u}_2 - u_2^*)]^2\big). \quad (21)$$

By equations (6) and (20), we have

$$\alpha_2 = -(\bar{u}_2 - u_2^*) - \frac{g_{xu}^2}{g_{uu}^2} f_u^1 \cdot (\bar{u}_1 - u_1^*) + O\big([(\bar{u}_1 - u_1^*) + (\bar{u}_2 - u_2^*)]^2\big), \quad (22)$$

$$\beta_2(x_2 - \bar{x}_2) = -\frac{g_{xu}^2}{g_{uu}^2}(x_2 - \bar{x}_2) = -\frac{g_{xu}^2}{g_{uu}^2}[f^1(\bar{x}_1, u_1) - f^1(\bar{x}_1, \bar{u}_1)]$$

$$= -\frac{g_{xu}^2}{g_{uu}^2} f_u^1 \cdot (u_1 - \bar{u}_1) + O\big((u_1 - \bar{u}_1)^2\big). \quad (23)$$

Since $u_2 = \bar{u}_2 + \alpha_2 + \beta_2 \cdot (x_2 - \bar{x}_2)$ (from equation $(A.1.11)$ in the Appendix), then

$$u_2 = \bar{u}_2 - (\bar{u}_2 - u_2^*) - \frac{g_{xu}^2}{g_{uu}^2} f_u^1 \cdot (\bar{u}_1 - u_1^*) - \frac{g_{xu}^2}{g_{uu}^2} f_u^1 \cdot (u_1 - \bar{u}_1)$$

$$+ O\big([(\bar{u}_1 - u_1^*) + (\bar{u}_2 - u_2^*)]^2 + (u_1 - \bar{u}_1)^2\big)$$

$$= u_2^* - \frac{g_{xu}^2}{g_{uu}^2} f_u^1 \cdot (u_1 - u_1^*) + O\big([(\bar{u}_1 - u_1^*) + (\bar{u}_2 - u_2^*)]^2 + (u_1 - \bar{u}_1)^2\big). \quad (24)$$

Thus, we have

$$u_2 - u_2^* = -\frac{g_{xu}^2}{g_{uu}^2} f_u^1 \cdot (u_1 - u_1^*) + O\big([(\bar{u}_1 - u_1^*) + (\bar{u}_2 - u_2^*)]^2 + (u_1 - \bar{u}_1)^2\big) \quad (25)$$

or

$$u_2 - u_2^* = -\frac{g_{xu}^2}{g_{uu}^2} f_u^1 \cdot (u_1 - u_1^*) + O\big([(\bar{u}_1 - u_1^*) + (\bar{u}_2 - u_2^*)]^2 + (u_1 - u_1^*)^2$$

$$+ (u_1 - u_1^*)(\bar{u}_1 - u_1^*)\big). \quad (26)$$

Substituting equation (20) into equations (9) and (10), we have

$$C_1 = \frac{1}{2}\{g^1_{uu} + (f^1_u)^2 \cdot [g^2_{xx} - \frac{(g^2_{xu})^2}{g^2_{uu}}] + g^2_x \cdot f^1_{uu}\} + O\big((\bar{u}_1 - u^*_1) + (\bar{u}_2 - u^*_2)\big), \tag{27}$$

$$D_1 = g^1_u + g^2_x \cdot f^1_u - g^2_{xu} \cdot f^1_u \cdot (\bar{u}_2 - u^*_2) - \frac{(g^2_{xu} \cdot f^1_u)^2}{g^2_{uu}}(\bar{u}_1 - u^*_1)$$

$$+ O\big([(\bar{u}_1 - u^*_1) + (\bar{u}_2 - u^*_2)]^2\big). \tag{28}$$

Substituting equation (21) into equation (28), we have

$$D_1 = (g^1_{uu} + g^2_{xx} \cdot (f^1_u)^2 + g^2_x \cdot f^1_{uu}) \cdot (\bar{u}_1 - u^*_1) + g^2_{xu} \cdot f^1_u \cdot (\bar{u}_2 - u^*_2)$$

$$- g^2_{xu} \cdot f^1_u \cdot (\bar{u}_2 - u^*_2) - \frac{(g^2_{xu} \cdot f^1_u)^2}{g^2_{uu}}(\bar{u}_1 - u^*_1) + O\big([(\bar{u}_1 - u^*_1) + (\bar{u}_2 - u^*_2)]^2\big). \tag{29}$$

Simplifying this, we obtain

$$D_1 = \{g^1_{uu} + (f^1_u)^2 \cdot [g^2_{xx} - \frac{(g^2_{xu})^2}{g^2_{uu}}] + g^2_x \cdot f^1_{uu}\} \cdot (\bar{u}_1 - u^*_1) + O\big([(\bar{u}_1 - u^*_1) + (\bar{u}_2 - u^*_2)]^2\big). \tag{30}$$

Since $\alpha_1 = -\frac{D_1}{2C_1}$, replacing C_1 by equation (27) and D_1 by equation (30), we have

$$\alpha_1 = -(\bar{u}_1 - u^*_1) + O\big([(\bar{u}_1 - u^*_1) + (\bar{u}_2 - u^*_2)]^2\big). \tag{31}$$

Thus, by $u_1 = \bar{u}_1 + \alpha_1$ (equation $(A.1.11)$ in the Appendix), we have

$$u_1 = u^*_1 + O\big([(\bar{u}_1 - u^*_1) + (\bar{u}_2 - u^*_2)]^2\big)$$

or

$$u_1 - u^*_1 = O\big([(\bar{u}_1 - u^*_1) + (\bar{u}_2 - u^*_2)]^2\big). \tag{32}$$

From equations (26) and (32), it follows that

$$u_1 - u^*_1 = O\big([(\bar{u}_1 - u^*_1) + (\bar{u}_2 - u^*_2)]^2\big), \tag{33}$$

$$u_2 - u^*_2 = O\big([(\bar{u}_1 - u^*_1) + (\bar{u}_2 - u^*_2)]^2\big). \tag{34}$$

So, we have

$$\|\vec{U}(2) - \vec{U}^*(2)\| \leq c(\bar{\vec{X}}(2), \bar{\vec{U}}(2)) \cdot \|\bar{\vec{U}}(2) - \vec{U}^*(2)\|^2 \tag{35}$$

where $c(\bar{\vec{X}}(2), \bar{\vec{U}}(2))$ is a positive function of $g^t,\ t = 1, 2,\ f^1$ and their derivatives.

From assumptions 1), 4) and 5) and from the continuity of $c(\bar{\vec{X}}(2), \bar{\vec{U}}(2))$, we know that $c(\bar{\vec{X}}(2), \bar{\vec{U}}(2))$ is bounded on D; hence there exists a constant $c > 0$ such that $c(\bar{\vec{X}}(2), \bar{\vec{U}}(2)) \leq c$ for all $(\bar{x}_t, \bar{u}_t) \in D,\ t = 1, 2$. By equation (35), we have

$$\|\vec{U}(2) - \vec{U}^*(2)\| \leq c \cdot \|\bar{\vec{U}}(2) - \vec{U}^*(2)\|^2. \tag{36}$$

This implies that the induction hypothesis is true for the case $N = 2$.

b) Suppose that the induction hypothesis is true for $N = k-1$, then there exists a constant $c_1 > 0$ such that

$$\|\vec{U}(k-1) - \vec{U}^*(k-1)\| \leq c_1 \cdot \|\bar{\vec{U}}(k-1) - \vec{U}^*(k)\|^2 \tag{37}$$

where $\vec{U}^*(k-1)$ is the optimal solution for the following $(k-1)$ time-step problem

$$\min_{u_1, \cdots, u_{k-1}} \sum_{t=1}^{k-1} g^t(x_t, u_t) \tag{38}$$

$$\text{where } x_{t+1} = f^t(x_t, u_t) \quad t = 1, \cdots, k-2,$$

$$x_1 \equiv \bar{x}_1 \text{ given and fixed.}$$

Below we will show that for the case $N = k$, our induction hypothesis is still true. In the case of $N = k$, our problem is the following

$$\min_{u_1, \cdots, u_k} \sum_{t=1}^{k} g^t(x_t, u_t) \tag{39}$$

$$\text{where } x_{t+1} = f^t(x_t, u_t) \quad t = 1, \cdots, k-1,$$

$$x_1 \equiv \bar{x}_1 \text{ given and fixed.}$$

The application of the DDP algorithm to problem (39) results in

$$A_k = \frac{1}{2} g_{xx}^k, \quad B_k = g_{xu}^k, \quad C_k = \frac{1}{2} g_{uu}^k, \quad D_k = g_u^k, \quad E_k = g_x^k, \tag{40}$$

(From assumption 3) and equation (40), we know $\quad C_k > 0 \quad$ or $\quad g_{uu}^k > 0.$)

$$P_k = \frac{1}{2}g_{xx}^k - \frac{1}{2}\frac{(g_{xu}^k)^2}{g_{uu}^k}, \qquad Q_k = -\frac{g_u^k \cdot g_{xu}^k}{g_{uu}^k} + g_x^k, \tag{41}$$

$$\alpha_k = -\frac{g_u^k}{g_{uu}^k}, \qquad \beta_k = -\frac{g_{xu}^k}{g_{uu}^k}, \tag{42}$$

$$A_{k-1} = \frac{1}{2}g_{xx}^{k-1} + (f_x^{k-1})^2 \cdot [\frac{1}{2}g_{xx}^k - \frac{1}{2}\frac{(g_{xu}^k)^2}{g_{uu}^k}] + \frac{1}{2}f_{xx}^{k-1} \cdot (-\frac{g_u^k \cdot g_{xu}^k}{g_{uu}^k} + g_x^k), \tag{43}$$

$$B_{k-1} = g_{xu}^{k-1} + 2(f_x^{k-1}) \cdot (f_u^{k-1}) \cdot [\frac{1}{2}g_{xx}^k - \frac{1}{2}\frac{(g_{xu}^k)^2}{g_{uu}^k}] + f_{xu}^{k-1} \cdot (-\frac{g_u^k \cdot g_{xu}^k}{g_{uu}^k} + g_x^k), \tag{44}$$

$$C_{k-1} = \frac{1}{2}g_{uu}^{k-1} + (f_u^{k-1})^2 \cdot [\frac{1}{2}g_{xx}^k - \frac{1}{2}\frac{(g_{xu}^k)^2}{g_{uu}^k}] + \frac{1}{2}f_{uu}^{k-1} \cdot (-\frac{g_u^k \cdot g_{xu}^k}{g_{uu}^k} + g_x^k), \tag{45}$$

$$D_{k-1} = g_u^{k-1} + f_u^{k-1} \cdot (-\frac{g_u^k \cdot g_{xu}^k}{g_{uu}^k} + g_x^k), \tag{46}$$

$$E_{k-1} = g_x^{k-1} + f_x^{k-1} \cdot (-\frac{g_u^k \cdot g_{xu}^k}{g_{uu}^k} + g_x^k). \tag{47}$$

Our approach here is to combine the last two stages (stage $k-1$ and stage k) in problem (39) together so that we can convert problem (39) into a $(k-1)$ time-step problem which is in the form of problem (38). Now, define

$$G(x_{k-1}, u_{k-1}) \doteq g^{k-1}(x_{k-1}, u_{k-1}) + g^k(x_k, \bar{u}_k + \alpha_k + \beta_k(x_k - \bar{x}_k)) \tag{48}$$

$$\begin{aligned} \text{where } \; x_k &= f^{k-1}(x_{k-1}, u_{k-1}), \\ \alpha_k &\doteq -g_u^k / g_{uu}^k, \\ \beta_k &\doteq -g_{xu}^k / g_{uu}^k, \end{aligned}$$

where all derivatives are evaluated at $\bar{x}_k$ and $\bar{u}_k$. Therefore, it is easy to see that $G(x,u)$ is in $C^2(R^{n+m})$.

By introducing function G, we can convert problem (39) into the following problem

$$\min_{u_1, \cdots, u_{k-1}} \{[\sum_{t=1}^{k-2} g^t(x_t, u_t)] + G(x_{k-1}, u_{k-1})\} \tag{49}$$

$$\text{where } x_{t+1} = f^t(x_t, u_t) \quad t = 1, \cdots, k-2,$$
$$x_1 \equiv \bar{x}_1 \text{ given and fixed.}$$

After applying the DDP algorithm to problem (49), we get $\hat{A}_{k-1}$, $\hat{B}_{k-1}$, $\hat{C}_{k-1}$, $\hat{D}_{k-1}$ and $\hat{E}_{k-1}$ as follows:

$$\hat{A}_{k-1} = \frac{1}{2}g_{xx}^{k-1} + (f_x^{k-1})^2 \cdot [\frac{1}{2}g_{xx}^k - \frac{1}{2}\frac{(g_{xu}^k)^2}{g_{uu}^k}] + \frac{1}{2}f_{xx}^{k-1} \cdot (-\frac{g_u^k \cdot g_{xu}^k}{g_{uu}^k} + g_x^k), \tag{50}$$

$$\hat{B}_{k-1} = g_{xu}^{k-1} + 2(f_x^{k-1}) \cdot (f_u^{k-1}) \cdot [\frac{1}{2}g_{xx}^k - \frac{1}{2}\frac{(g_{xu}^k)^2}{g_{uu}^k}] + f_{xu}^{k-1} \cdot (-\frac{g_u^k \cdot g_{xu}^k}{g_{uu}^k} + g_x^k), \tag{51}$$

$$\hat{C}_{k-1} = \frac{1}{2}g_{uu}^{k-1} + (f_u^{k-1})^2 \cdot [\frac{1}{2}g_{xx}^k - \frac{1}{2}\frac{(g_{xu}^k)^2}{g_{uu}^k}] + \frac{1}{2}f_{uu}^{k-1} \cdot (-\frac{g_u^k \cdot g_{xu}^k}{g_{uu}^k} + g_x^k), \tag{52}$$

$$\hat{D}_{k-1} = g_u^{k-1} + f_u^{k-1} \cdot (-\frac{g_u^k \cdot g_{xu}^k}{g_{uu}^k} + g_x^k), \tag{53}$$

$$\hat{E}_{k-1} = g_x^{k-1} + f_x^{k-1} \cdot (-\frac{g_u^k \cdot g_{xu}^k}{g_{uu}^k} + g_x^k). \tag{54}$$

Then, from equations (43) – (47) and equations (50) – (54), it can be seen that

$$\hat{A}_{k-1} = A_{k-1},\ \hat{B}_{k-1} = B_{k-1},\ \hat{C}_{k-1} = C_{k-1},\ \hat{D}_{k-1} = D_{k-1},\ \hat{E}_{k-1} = E_{k-1}. \tag{55}$$

Assumption 3) indicates that C_{k-1} is positive definite. Equation (55) implies that $\hat{C}_{k-1}$ is also positive definite.

Since the DDP algorithm is a backward recursive method, from equation (55), and the fact that the objective functions and transition equations are identical for $t = 1, \cdots, k-2$, in problems (38) and (49), it follows that

$$\hat{A}_t = A_t,\ \hat{B}_t = B_t,\ \hat{C}_t = C_t,\ \hat{D}_t = D_t,\ \hat{E}_t = E_t \ \text{ for } t = 1, \cdots, k-1. \tag{56}$$

Therefore, equation (56) and the definition of the function G indicates that problem (39) and problem (49) generate the same sequence of $\vec{U}(k)$, where $u_k = \bar{u}_k + \alpha_k + \beta_k \cdot \Big(f^{k-1}(x_{k-1}, u_{k-1}) - f^{k-1}(\bar{x}_{k-1}, \bar{u}_{k-1})\Big)$ for problem (49). Since the application of the DDP algorithm to problem (39) at last stage is just Newton's method, then

$$|u_k - u_k^*| \le c_2 \cdot |\bar{u}_k - u_k^*|^2 \qquad \text{for some } c_2 > 0. \tag{57}$$

Combining equations (37) and (57) together, we have

$$\|\vec{U}(k) - \vec{U}^*(k)\| \le c(\bar{\vec{X}}(k), \bar{\vec{U}}(k)) \cdot \|\bar{\vec{U}}(k) - \vec{U}^*(k)\|^2 \tag{58}$$

where $c(\bar{\vec{X}}(k), \bar{\vec{U}}(k))$ is a positive function of g^t, $t = 1, \cdots, k$, f^t, $t = 1, \cdots, k-1$ and their derivatives.

Following the same reason as in the case $N = 2$ (equation (36)), there exists a constant $c > 0$ such that $c(\vec{X}(k), \vec{U}(k)) \le c$ for all $(x_t, u_t) \in D$, $t = 1, \cdots, k$. This proves the induction hypothesis for the case $N = k$. Therefore, the conclusion of this theorem is true for all values of N. ∎

Appendix The Unconstrained Differential Dynamic Programming Algorithm

In this appendix, we summarize the unconstrained DDP algorithm described by Yakowitz and Rutherford [8] for solving problem (P_0).

Algorithm

Step 1: (Initialize parameters and compute loss and trajectory associated with the given policy)

i) Set $P_{N+1} = 0_{n\times n}$, $Q_{N+1} = 0_n$, $\theta_{N+1} = 0$.

ii) $\bar{x}_{t+1} = f(\bar{x}_t, \bar{u}_t, t)$, $\quad t = 1, \ldots, N-1$, $\quad \bar{x}_1 \equiv \bar{x}_1$.

iii) $J(\bar{\vec{U}}) = \sum_{t=1}^{N} g(\bar{x}_t, \bar{u}_t, t)$.

Step 2: (Perform the backward sweep: Perform steps i) to iv) below, recursively, for $t = N, \ldots, 1$)

i) compute A_t, B_t, C_t, D_t and E_t according to

$$A_t = \frac{1}{2}[g_{xx} + 2(\frac{\partial f}{\partial x})^T P_{t+1}(\frac{\partial f}{\partial x}) + \sum_{i=1}^{n}(Q_{t+1})_i(f_i)_{xx}], \tag{A.1.1}$$

$$B_t^T = g_{xu} + 2(\frac{\partial f}{\partial x})^T P_{t+1}(\frac{\partial f}{\partial u}) + \sum_{i=1}^{n}(Q_{t+1})_i(f_i)_{xu}, \tag{A.1.2}$$

$$C_t = \frac{1}{2}[g_{uu} + 2(\frac{\partial f}{\partial u})^T P_{t+1}(\frac{\partial f}{\partial u}) + \sum_{i=1}^{n}(Q_{t+1})_i(f_i)_{uu}], \tag{A.1.3}$$

$$D_t = g_u + (\frac{\partial f}{\partial u})^T Q_{t+1}, \tag{A.1.4}$$

$$E_t = g_x + (\frac{\partial f}{\partial x})^T Q_{t+1}. \tag{A.1.5}$$

ii) compute P_t and Q_t according to

$$P_t = A_t - \tfrac{1}{4} B_t^T C_t^{-1} B_t, \tag{A.1.6}$$

$$Q_t = -\tfrac{1}{2} D_t^T C_t^{-1} B_t + E_t, \tag{A.1.7}$$

store P_t and Q_t in memory, replacing P_{t+1} and Q_{t+1}.

iii) Compute α_t and β_t according to

$$\alpha_t = -\tfrac{1}{2} C_t^{-1} D_t, \tag{A.1.8}$$

$$\beta_t = -\tfrac{1}{2} C_t^{-1} B_t \tag{A.1.9}$$

and store in memory.

iv) Compute

$$\theta_t = -\tfrac{1}{2} D_t^T C_t^{-1} D_t + \theta_{t+1} \tag{A.1.10}$$

store θ_t in place of θ_{t+1} in memory.

Step 3: (Compute the successor policy)

i) Set $\varepsilon = 1$.

ii) Compute $u_t(\varepsilon)$ recursively for $t = 1, \ldots, N$ according to

$$u_t(\varepsilon) = \varepsilon \alpha_t + \beta_t (x_t - \bar{x}_t) + \bar{u}_t \ , \tag{A.1.11}$$

$$x_{t+1} = f(x_t, u_t(\varepsilon), t). \tag{A.1.12}$$

iii) Compute $J(\vec{U}(\varepsilon)) = \sum_{t=1}^{N} \ g(x_t, u_t(\varepsilon), t)$.

iv) If $J(\vec{U}(\varepsilon)) - J(\bar{\vec{U}}) < \varepsilon \dfrac{\theta_1}{2}$, set $\bar{\vec{U}} = \vec{U}(\varepsilon)$ and go to step 1;

otherwise set $\varepsilon = \varepsilon/2$, go to step 3 ii).

References

[1] D. Jacobson and D. Mayne, *Differential Dynamic Programming*, Elsevier Sci. Publ., New York, NY, 1970.

[2] L-Z. Liao, "Numerically Efficient Algorithms for Unconstrained and Constrained Differential Dynamic Programming in Discrete-Time, Nonlinear Systems," Ph. D. Thesis, Cornell University, 1990.

[3] L-Z. Liao and C. A. Shoemaker, "Convergence in unconstrained discrete-time differential dynamic programming," (*IEEE Trans. Automat. Contr.*) (in press).

[4] D. Mayne, "A second-order gradient method for determining optimal trajectories of non-linear discrete-time systems," *Intnl. J. Control*, Vol. 3, pp. 85-95, 1966.

[5] D. M. Murray and S. Yakowitz, "Differential dynamic programming and Newton's method for discrete optimal control problems," *JOTA*, Vol. 43, pp. 395-414, 1984.

[6] J. F. A. De O. Pantoja, "Algorithm for constrained optimization problems," Ph. D. Thesis, Imperial College of Science and Technology, University of London, 1983.

[7] S. Yakowitz, "Theoretical and computational advances in differential dynamic programming," *Control and Cybernetics*, Vol. 17, No. 2-3, pp. 173-189, 1988.

[8] S. Yakowitz and B. Rutherford, "Computational aspects of discrete-time optimal control," *Appl. Math. Comput.*, Vol. 15, pp. 29-45, 1984.