REVERSED CONTROL PROCESSES

Iwamoto, Seiichi
Department of Mathematics, Faculty of Integrated Arts and Sciences, Hiroshima University

https://doi.org/10.5109/13135

出版情報：統計数理研究. 19 (1/2), pp.1-11, 1980-03. Research Association of Statistical Sciences
バージョン：
権利関係：

REVERSED CONTROL PROCESSES

By

Seiichi IWAMOTO*

(Received January 5, 1979)

Abstract

A reversal operation for dynamic programming (DP) transforms a (main) control process to a reversed control process. It is applied to the well known linear equation and quadratic criterion control process and to a terminal control process. Not solving the recursive equation directly, but merely applying the author's Reverse Theorem in DP, we can automatically obtain the solution of the recursive equation for reversed process.

1. Introduction

Recently the author has proposed, among other operations, a reversal operation for dynamic programming (DP) on one-dimensional state space. He [6] has established Reverse Theorem on general state DPs. These DPs are more strictly defined than those of Bellman's [1]. Therefore, the class of reversible DPs may be narrow. However, it includes an important deteministic linear equation and quadratic criterion control process and other terminal control processes.

The paper applies the author's Reverse Theorem to a discrete time finite horizon control process in § 4 and to a particular terminal optimization problem in § 5. We generalizes some standard control processes in Bellman [2, p. 329; 3, p. 116, p. 193; 4, p. 20] to a nonstationary process. The latter is a modified version of the optimization problem [2, p. 305].

In § 2, we specify a (main) DP by an ordered seven-tuple satisfying some algebraically regular conditions. These conditions make the main DP reversible. A reversed DP is defined by negation of optimizer, reversion of time (consequently reversion of state spaces and reward spaces, respectively), inversion of reward functions, and substitution of optimal reward function for terminal reward function (§ 3). Thus the revesed DP of the revesed DP is an original main DP. We state Reverse Theorem between main DP and reversed DP without proof: A pair of optimal reward functions and an optimal policy for main DP characterizes a pair of them for reversed DP in a reverse sense.

In § 4 and § 5 we introduce "reversed control processes" for only two cases. However, the term "reversed" posseses wide applicabilities to more general control

* Department of Mathematics, Faculty of Integrated Arts and Sciences, Hiroshima University, Hiroshima 730, Japan.

processes by relaxing the regularity conditions stated above.

2. Reversible dynamic programming

Let S and S' be two nonempty sets. Then $\mathcal{G}(S, S')$ denotes the set of all one-to-one mappings from S onto S'. We remark that $T \in \mathcal{G}(S, S')$ if and only if $T^{-1} \in \mathcal{G}(S', S)$, where T^{-1} is the inverse mapping of T. Let R and R' be two intervals of the one-dimensional Euclidean space R^1. Then $\mathcal{F}(R, R')$ denotes the set of all strictly increasing functions from R onto R'. Note that any function f in $\mathcal{F}(R, R')$ is continuous and therefore it yields a homeomorphism from R onto R'. Further, $f \in \mathcal{F}(R, R')$ if and only if $f^{-1} \in \mathcal{F}(R', R)$. We also have $\mathcal{F}(R, R') \subset \mathcal{G}(R, R')$. Given a two-variable mapping $h : A \times B \to C$ we define two one-variable mappings $h_a : B \to C$ and $h_b : A \to C$ by

$$h^a(b) = h(a, b), \qquad h_b(a) = h(a, b),$$

respectively.

A *reversible dynamic programming* (*DP*) $\mathcal{D}$ is specified by an ordered seven-tuple (Opt, $\{S_n\}_1^{N+1}$, $\{R_n\}_1^{N+1}$, $\{A_n\}_1^N$, $\{f_n\}_1^N$, k, $\{T_n\}_1^N$) with the following meanings:

(i) N is a positive integer, the *number of stages*.

(ii) $\{S_n\}_1^{N+1}$ are nonempty sets satisfying $\mathcal{G}(S_n, S_{n+1}) \neq \phi$ for $1 \leqq n \leqq N$. S_n and its element s_n are called the *n-th state space* and *n-th state*, respectively.

(iii) $\{R_n\}_1^{N+1}$ are intervals of R^1 satisfying $\mathcal{F}(R_{n+1}, R_n) \neq \phi$ for $1 \leqq n \leqq N$. R_n and its element r_n are called the *n-th reward space* and *n-th reward*, respectively.

(iv) A_n is a nonempty set, the *n-th action space*, whose element a_n is called the *n-th action*.

(v) $f_n : S_n \times A_n \times S_{n+1} \to \mathcal{F}(R_{n+1}, R_n)$ is the *n-th reward function*. We frequently write $f_n(s_n, a_n, s_{n+1})(r_{n+1})$, $f_n(s_n, a_n, s_{n+1}, r_{n+1})$ or $f_n^{(s_n, a_n, s_{n+1})}(r_{n+1})$ for $(s_n, a_n, s_{n+1}, r_{n+1}) \in S_n \times A_n \times S_{n+1} \times R_{n+1}$.

(vi) $k : S_{N+1} \to R_{N+1}$ is the *terminal reward function*.

(vii) $T_n : A_n \to \mathcal{G}(S_n, S_{n+1})$ is the *n-th state transformation*. We frequently write $T_n(a_n)(s_n)$, $T_n(s_n, a_n)$ or $T_{na_n}(s_n)$ for $(s_n, a_n) \in S_n \times A_n$.

(viii) Opt denotes either Max or Min, the *optimizer*. According as Opt=Max or Min, it represents the following optimization (either maximization or minimization) problem:

Optimize

$$f_1(s_1, a_1, s_2)(f_2(s_2, a_2, s_3)(\cdots(f_N(s_N, a_N, s_{N+1})(k(s_{N+1})))\cdots))$$

subject to

$$\text{(i)} \qquad T_n(a_n)(s_n) = s_{n+1} \qquad 1 \leqq n \leqq N$$

$$\text{(ii)} \qquad a_n \in A_n \qquad 1 \leqq n \leqq N.$$

More symbolically this problem may be written as follows:

Optimize

$$f_1^{(s_1, a_1, s_2)} \circ f_2^{(s_2, a_2, s_3)} \circ \cdots f_N^{(s_N, a_N, s_{N+1})} \circ k(s_{N+1})$$

subject to

$$\text{(i)} \qquad T_{na_n}(s_n)=s_{n+1} \qquad 1\leqq n\leqq N$$

$$\text{(ii)} \qquad a_n\in A_n \qquad 1\leqq n\leqq N.$$

Further it may also be expressed in a standard from:

Optimize

$$f_1(s_1,\ a_1,\ s_2,\ f_2(s_2,\ a_2,\ s_3,\ \cdots,\ f_N(s_N,\ a_N,\ s_{N+1},\ k(s_{N+1}))\cdots))$$

(2.1) subject to

$$\text{(i)} \qquad T_n(s_n,\ a_n)=s_{N+1} \qquad 1\leqq n\leqq N$$

$$\text{(ii)} \qquad a_n\in A_n \qquad 1\leqq n\leqq N.$$

We call $\mathcal{D}$ the *main DP*. First it starts at a given initial state $s_1\in S_1$, a first action $a_1\in A_1$ is chosen, and it goes to the second state $s_2=T_1(s_1,\ a_1)\in S_2$. A second action $a_2\in A_2$ moves the main DP $\mathcal{D}$ to the third state $s_3=T_2(s_2,\ a_2)\in S_3$, and so on. Finally $\mathcal{D}$ terminates at the terminal state $s_{N+1}=T_N(s_N,\ a_N)\in S_{N+1}$. The experience starting at s_1 and choosing a sequence $\{a_1,\ a_2,\ \cdots,\ a_N\}$ yields the 'total' reward

$$f_1^{(s_1,a_1,s_2)}\circ f_2^{(s_2,a_2,s_3)}\circ\cdots\circ f_N^{(s_N,a_N,s_{N+1})}\circ k(s_{N+1}).$$

The main DP $\mathcal{D}$ itself wishes to optimize this total reward subject to the state-action constraint.

Let us now consider the $(N-n+1)$-*subproblem* of (2.1) as follows:

Optimize

$$f_n(s_n,\ a_n,\ s_{n+1},\ f_{n+1}(s_{n+1},\ a_{n+1},\ s_{n+2},\ \cdots,\ f_N(s_N,\ a_N,\ s_{N+1},\ k(s_{N+1}))\cdots))$$

(2.2) subject to

$$\text{(i)} \qquad T_m(s_m,\ a_m)=s_{m+1} \qquad n\leqq m\leqq M$$

$$\text{(ii)} \qquad a_m\in A_m \qquad n\leqq m\leqq M,$$

where $s_n\in S_n$, $1\leqq n\leqq N$. The optimum value function of the $(N-n+1)$-subproblem is denoted by $u^{N-n+1}\colon S_n\rightarrow R_n$ if it exists:

$$u^{N-n+1}(s_n)=\mathop{\mathrm{Opt}}_{\substack{T_{m a_m}(s_m)=s_{m+1}\\ a_m\in A_m\\ n\leqq m\leqq M}} f_n^{(s_n,a_n,s_{n+1})}\circ f_{n+1}^{(s_{n+1},a_{n+1},s_{n+2})}\circ\cdots\circ f_N^{(s_N,a_N,s_{N+1})}\circ k(s_{N+1})$$

$$s_n\in S_n, \qquad 1\leqq n\leqq N.$$

Further, we define $u^0\colon S_{N+1}\rightarrow R_{N+1}$ by

$$u^0(s_{N+1})=k(s_{N+1}) \qquad s_{N+1}\in S_{N+1}.$$

The functions $\{u^0,\ u^1,\ \cdots,\ u^N\}$ are called *optimal reward functions* for main DP $\mathcal{D}$. Two adjacent optimum value functions are combined as follows:

THEOREM. 1. (RECURSIVE FORMULA FOR $\mathscr{D}$)

$$u^{N-n+1}(s_n)=\operatorname*{Opt}_{a_n\in A_n} f_n(s_n,\ a_n,\ T_n(s_n,\ a_n),\ u^{N-n}(T_n(s_n,\ a_n))) \qquad s_n\in S_n,\quad 1\leqq n\leqq N \tag{2.3}$$

$$u^0(s_{N+1})=k(s_{N+1}) \qquad s_{N+1}\in S_{N+1}.$$

PROOF. Easy.

A *policy* of $\mathscr{D}$ is a finite sequence $\{\pi_1, \pi_2, \cdots, \pi_N\}$ of mappings $\pi_n: S_n\to A_n$ for $1\leqq n\leqq N$. A policy $\{\pi_1^*, \pi_2^*, \cdots, \pi_N^*\}$ is called *optimal* for $\mathscr{D}$ if for each $s_n\in S_n$, $1\leqq n\leqq N$ the $\pi_n^*(s_n)$ attains the optimum value of (2.3).

3. Reversed dynamic programming and Reverse Theorem

When the main DP $\mathscr{D}$ has an optimum value function $u^N: S_1\to R_1$, a *reversed DP* $\mathscr{D}_{-1}$ is specified by the ordered seven-tuple $(\overline{\text{Opt}}, \{S'_n\}_1^{N+1}, \{R'_n\}_1^{N+1}, \{A'_n\}_1^N, \{f'_n\}_1^N, k', \{T'_n\}_1^N)$ where

$$\overline{\text{Opt}}=\text{Min} \quad \text{if} \quad \text{Opt}=\text{Max}$$

$$=\text{Max} \quad \text{if} \quad \text{Opt}=\text{Min}$$

$$S'_n=S_{N-n+2}, \qquad R'_n=R_{N-n+2}, \qquad A'_n=A_{N-n+1}$$

$$(s'_n=s_{N-n+2}, \qquad r'_n=r_{N-n+2}, \qquad a'_n=a_{N-n+1})$$

$$f'_n(s'_{n'}, a'_{n'}, s'_{n+1})=(f_{N-n+1}(s_{N-n+1}, a_{N-n+1}, s_{N-n+2}))^{-1}$$

$$T'_n(a'_n)=(T_{N-n+1}(a_{N-n+1}))^{-1}$$

$$k'=u^N.$$

Starting backwards at state $s'_1=s_{N+1}\in S_{N+1}=S'_1$ and terminating at state $s'_{N+1}=s_1\in S_1=S'_{N+1}$, the reversed DP $\mathscr{D}_{-1}$ represents the problem:

$\overline{\text{Optimize}}$

$$f'_1(s'_1,\ a'_1,\ s'_2,\ f'_2(s'_2,\ a'_2,\ s'_3,\ \cdots,\ f'_N(s'_N,\ a'_N,\ s'_{N+1},\ k'(s'_{N+1}))\cdots))$$

(3.1) subject to

(i) $\qquad T'_n(s'_n,\ a'_n)=s'_{n+1} \qquad 1\leqq n\leqq N$

(ii) $\qquad a'_n\in A'_n \qquad 1\leqq n\leqq N.$

Rewriting in terms of components of the main DP $\mathscr{D}$, this problem reduces to

$\overline{\text{Optimize}}$

$$(f_N^{(s_N, a_N, s_{N+1})})^{-1}\circ(f_{N-1}^{(s_{N-1}, a_{N-1}, s_N)})^{-1}\circ\cdots(f_1^{(s_1, a_1, s_2)})^{-1}\circ u^N(s_1)$$

subject to

$$\text{(i)} \qquad (T_{na_n})^{-1}(s_{n+1})=s_n \qquad N\geqq n\geqq 1$$

$$\text{(ii)} \qquad a_n\in A_n \qquad N\geqq n\geqq 1\,,$$

where $N\geqq n\geqq 1$ means that the time n runs backwards $N+1, N, \cdots, 2, 1$. Thus, we may specify the reversed DP $\mathcal{D}_{-1}$ by

$$(\overline{\mathrm{Opt}},\ \{S_n\}_{N+1}^1,\ \{R_n\}_{N+1}^1,\ \{A_n\}_N^1,\ \{(f_n^{(s_n, a_n, s_{n+1})})^{-1}\}_N^1,\ u^N,\ \{(T_{na_n})^{-1}\}_N^1)\,.$$

The $(N-n+1)$-*subproblem* of (3.1) becomes as follows:
$\overline{\text{Optimize}}$

$$f'_n(s'_n,\ a'_n,\ s'_{n+1},\ f'_{n+1}(s'_{n+1},\ a'_{n+1},\ s'_{n+2},\ \cdots,\ f'_N(s'_N,\ a'_N,\ s'_{N+1},\ k'(s'_{N+1}))\cdots))$$

subject to

$$\text{(i)} \qquad T'_m(s'_m,\ a'_m)=s'_{m+1} \qquad n\leqq m\leqq N$$

$$\text{(ii)} \qquad a'_m\in A'_m \qquad n\leqq m\leqq N$$

where $s'_n\in S'_n$, $1\leqq n\leqq N$. The $\overline{\text{optimum}}$ value function of this subproblem is denoted by $v^{N-n+1}: S'_n\to R'_n$ or $v^{N-n+1}: S_{N-n+2}\to R_{N-n+2}$ if it exists:

$$v^{N-n+1}(s_{N-n+2})=\overline{\mathrm{Opt}}_{\substack{(T_{ma_m})^{-1}(s_{m+1})=s_m\\ a_m\in A_m\\ N-n+1\geqq m\geqq 1}}(f_{N-n+1}^{(s_{N-n+1}, a_{N-n+1}, s_{N-n+2})})^{-1}\circ(f_{N-n}^{(s_{N-n}, a_{N-n}, s_{N-n+1})})^{-1}$$

$$\circ\cdots\circ(f_1^{(s_1, a_1, s_2)})^{-1}\circ u^N(s_1)$$

$$s_{N-n+2}\in S_{N-n+2}\,, \qquad 1\leqq n\leqq N\,.$$

Further we define $v^0: S_1\to R_1$ by

$$v^0(s_1)=u^N(s_1) \qquad s_1\in S_1\,.$$

The functions $\{v^0, v^1, \cdots, v^N\}$ are called *optimal reward functions* for reversed DP $\mathcal{D}_{-1}$. We have the following recursive relation:

THEOREM 2. (RECURSIVE FORMULA FOR $\mathcal{D}_{-1}$)

$$v^{N-n+1}(s_{N-n+2})=\overline{\mathrm{Opt}}_{a_{N-n+1}\in A_{N-n+1}}(f_{N-n+1}^{((T_{N-n+1}, a_{N-n+1})^{-1}(s_{N-n+2}), a_{N-n+1}, s_{N-n+2})})^{-1}$$

$$\circ v^{N-n}((T_{N-n+1},\ a_{N-n+1})^{-1}(s_{N-n+2}))$$

$$s_{N-n+2}\in S_{N-n+2}\,, \qquad 1\leqq n\leqq N$$

$$v^0(s_1)=u^N(s_1) \qquad s_1\in S_1\,.$$

PROOF. Easy.

Note that a policy of $\mathcal{D}_{-1}$ is a sequence $\{\sigma_1, \sigma_2, \cdots, \sigma_N\}$ of mappings $\sigma_n: S_{N-n+2}\to A_{N-n+1}$ for $1\leqq n\leqq N$. Let $U_n\in\mathcal{G}(S_{n+1}, S_n)$ for $1\leqq n\leqq N$. Then, clearly, $\{\pi_1, \pi_2, \cdots, \pi_N\}$ is a policy of $\mathcal{D}$ if and only if $\{\pi_N\circ U_N, \pi_{N-1}\circ U_{N-1}, \cdots, \pi_1\circ U_1\}$ is a policy of $\mathcal{D}_{-1}$. We define $T_{n\pi_n}: S_n\to S_{n+1}$ by

$$T_{n\pi_n}(s_n)=T_{n\pi_n(s_n)}(s_n) \qquad s_n\in S_n .$$

Our fundamental theorem is REVERSE THEOREM in dynamic programming:

THEOREM 3. (REVERSE THEOREM) (i) *The main DP* $\mathcal{D}$ *has optimal reward functions* $\{u^0, u^1, \cdots, u^N\}$ *and an optimal policy* $\{\pi_1^*, \pi_2^*, \cdots, \pi_N^*\}$ *if and only if the reversed DP* $\mathcal{D}_{-1}$ *has optimal reward functions* $\{u^N, u^{N-1}, \cdots, u^0\}$ *and an optimal policy* $\{\pi_N^*\circ(T_{N\pi_N^*})^{-1}, \pi_{N-1}^*\circ(T_{N-1,\pi_{N-1}^*})^{-1}, \cdots, \pi_1^*\circ(T_{1\pi_1^*})^{-1}\}$, *provided that* $T_{n\pi_n^*}\in\mathcal{G}(S_n, S_{n+1})$ *for* $1\leqq n\leqq N$.

(ii) *The reversed DP* $\mathcal{D}_{-1}$ *has optimal reward functions* $\{v^0, v^1, \cdots, v^N\}$ *and an optimal policy* $\{\hat{\sigma}_1, \hat{\sigma}_2, \cdots, \hat{\sigma}_N\}$ *if and only if the main DP* $\mathcal{D}$ *has optimal reward functions* $\{v^N, v^{N-1}, \cdots, v^0\}$ *and an optimal policy* $\{\hat{\sigma}_N\circ T'_{N\hat{\sigma}_N})^{-1}, \hat{\sigma}_{N-1}\circ(T'_{N,\hat{\sigma}_{N-1}})^{-1}, \cdots, \hat{\sigma}_1\circ(T'_{1\hat{\sigma}_1})^{-1}\}$, *provided that* $T'_{n\hat{\sigma}_n}\in\mathcal{G}(S_{N-n+2}, S_{N-n+1})$ *for* $1\leqq n\leqq N$.

PROOF. See REVERSE THEOREM in [6].

4. A linear equation quadratic criterion control process

For given two symmetric matrices A and B, let $A\geqq B$ ($A>B$) denote that $A-B$ is non-negative (positive) definite. (,) denotes the inner product.

Let p and N be positive integers. Let A_n $(1\leqq n\leqq N)$, Q_n $(1\leqq n\leqq N)$, K, R_n $(1\leqq n\leqq N)$ and B_n $(1\leqq n\leqq N)$ be $p\times p$-matrices. We assume that A_n and B_n are non-singular and that

$$Q_n\geqq 0, \qquad R_n>0, \qquad K>0 \tag{4.1}$$

where 0 is the matrix of all zero elements.

Let us now consider a more general nonstationary discrete control process. The problem is to minimize a total cost by addition of the individual costs

$$\sum_{n=1}^{N}[(x_n, Q_nx_n)+(y_n, R_ny_n)]+(x_{N+1}, Kx_{N+1})$$

subject to x_n and y_n related by the difference equation

$$x_{n+1}=A_nx_n+B_ny_n \qquad 1\leqq n\leqq N, \quad x_1=c$$

$$x_n\in R^p, \qquad y_n\in R^p \qquad 1\leqq n\leqq N.$$

This control process is represented by an N-stage main DP $\mathcal{D}=(\text{Min}, \{S_n\}_1^{N+1}, \{R_n\}_1^{N+1}, \{A_n\}_1^N, \{f_n\}_1^N, k, \{T_n\}_1^N)$ starting at an initial state $s_1=c$, where

$$S_n=R^p, \qquad R_n=R^1, \qquad A_n=R^p,$$

$$f_n(s_n, a_n, s_{n+1}, r_{n+1})=(s_n, Q_ns_n)+(a_n, R_na_n)+r_{n+1},$$

$$k(s_{N+1})=(s_{N+1}, Ks_{N+1}), \qquad T_n(s_n, a_n)=A_ns_n+B_na_n .$$

We call $\mathcal{D}$ the *main control process.* The main control process $\mathcal{D}$ represents the problem:

Minimize

$$\sum_{n=1}^{N}[(s_n,\ Q_n s_n)+(a_n,\ R_n a_n)]+(s_{N+1},\ K s_{N+1})$$

subject to

$$\text{(i)} \qquad A_n s_n + B_n a_n = s_{n+1} \qquad 1 \leqq n \leqq N$$

$$\text{(ii)} \qquad a_n \in R^p \qquad 1 \leqq n \leqq N.$$

We follow Bellman's analysis [2, p. 329; 3, p. 193]. First

$$u^0(s_{N+1}) = k(s_{N+1})$$

$$= (s_{N+1},\ K_{N+1} s_{N+1})$$

where

$$K_{N+1} = K.$$

Second we have

$$u^1(s_N) = \underset{a_N \in R^p}{\mathrm{Min}}[(s_N,\ Q_N s_N)+(a_N,\ R_N a_N)+u^0(A_N s_N + B_N a_N)]$$

$$= \underset{a_N \in R^p}{\mathrm{Min}}[(s_N,\ Q_N s_N)+(a_N,\ R_N a_N)+(A_N s_N + B_N a_N,\ K_{N+1}(A_N s_N + B_N a_N))].$$

By (4.1) the minimum value is attained by a_N, denoted by $\pi_N^*(s_N)$, satisfying

$$\frac{d}{da_N}[(s_N,\ Q_N s_N)+(a_N,\ R_N a_N)+(A_N s_N + B_N a_N,\ K_{N+1}(A_N s_N + B_N a_N))]=0$$

namely

$$\text{(4.2)} \qquad R_N a_N + B_N' K_{N+1}(A_N s_N + B_N a_N) = 0$$

Since $R_N + B_N' K_{N+1} B_N > 0$, we obtain

$$\pi_N^*(s_N) = L_N s_N$$

where

$$L_N = -(R_N + B_N' K_{N+1} B_N)^{-1} B_N' K_{N+1} A_N.$$

This $\pi_N^*(s_N)$ together with the identity (4.2) yields

$$u^1(s_N) = (s_N,\ K_N s_N)$$

where

$$K_N = Q_N + A_N'[K_{N+1}(A_N - B_N(R_N + B_N' K_{N+1} B_N)^{-1} B_N' K_{N+1} A_N)]$$

$$= Q_N + A_N'\{K_{N+1} - K_{N+1} B_N (R_N + B_N' K_{N+1} B_N)^{-1} B_N' K_{N+1}\} A_N$$

$$= Q_N + A_N'(K_{N+1}^{-1} + B_N R_N^{-1} B_N')^{-1} A_N$$

$$> 0.$$

In general the main control process $\mathscr{D}$ has quadratic optimal reward functions $\{u^0, u^1, \cdots, u^N\}$ and a linear optimal policy $\{\pi_1^*, \pi_2^*, \cdots, \pi_N^*\}$:

$$u^{N-n+1}(s_n) = (s_n,\ K_n s_n)$$

$$\pi_n^*(s_n) = L_n s_n$$

where

$$
(4.3)\quad
\begin{aligned}
&K_{N+1}=K\\
&K_n=Q_n+A'_n\{K_{n+1}-K_{n+1}B_n(R_n+B'_nK_{n+1}B_n)^{-1}B'_nK_{n+1}\}A_n\\
&\quad=Q_n+A'_n(K_{n+1}^{-1}+B_nR_n^{-1}B'_n)^{-1}A_n \qquad N\geqq n\geqq 1\\
&L_n=-(R_n+B'_nK_{n+1}B_n)^{-1}B'_nK_{n+1}A_n \qquad N\geqq n\geqq 1\,.
\end{aligned}
$$

Note that $A_1=\cdots=A_N$, $B_1=\cdots=B_N$, $Q_1\geqq\cdots\geqq Q_N$, $R_1\geqq\cdots\geqq R_N$ and $K_{N+1}<K_N$ imply

$$0<K_{N+1}<K_N<\cdots<K_2<K_1\,.$$

On the other hand, let us consider another control process. The problem is in turn to maximize a net gain by subtraction of the individual costs from the total cost

$$(x_1,\ K_1x_1)-\sum_{n=1}^{N}[(x_n,\ Q_nx_n)+(y_n,\ R_ny_n)]$$

subject to x_n and y_n related by the difference equation

$$
\begin{aligned}
&x_n=A_n^{-1}(x_{n+1}-B_ny_n) \qquad N\geqq n\geqq 1\,, \qquad x_{N+1}=c\\
&x_n\in R^p\,, \qquad y_n\in R^p \qquad\qquad N\geqq n\geqq 1
\end{aligned}
$$

where K_1 is determined backwards by (4.3) with initial condition $K_{N+1}=K$. This maximization problem can be represented by the reversed DP $\mathscr{D}_{-1}=(\text{Max},\ \{S_n\}_{N+1}^1,\ \{R_n\}_{N+1}^1,\ \{A_n\}_N^1,\ \{(f_n^{(s_n,a_n,s_{n+1})})^{-1}\}_N^1,\ u^N,\ \{(T_{na_n})^{-1}\}_N^1)$ starting at 'terminal' state $s_{N+1}=c$, where

$$
\begin{aligned}
&(f_n^{(s_n,a_n,s_{n+1})})^{-1}(r_n)=r_n-(s_n,\ Q_ns_n)-(a_n,\ R_na_n)\\
&u^N(s_1)=(s_1,\ K_1s_1)\\
&(T_{na_n})^{-1}(s_{n+1})=A_n^{-1}(s_{n+1}-B_na_n)\,.
\end{aligned}
$$

We call this reversed DP the *reversed control process.* Thus the reversed control process $\mathscr{D}_{-1}$ represents the problem:

Maximize

$$-\sum_{n=N}^{1}[(s_n,\ Q_ns_n)+(a_n,\ R_na_n)]+(s_1,\ K_1s_1)$$

subject to

(i) $\quad A_n^{-1}(s_{n+1}-B_na_n)=s_n \qquad N\geqq n\geqq 1$

(ii) $\quad a_n\in R^p \qquad N\geqq n\geqq 1\,.$

The problem is also written in the following forward process:

Maximize

$$\sum_{n=1}^{N}[(-(y_n,\ R_{N-n+1}y_n)-(x_{n+1},\ Q_{N-n+1}x_{n+1})]+(x_{N+1},\ K_1x_{N+1})$$

subject to

$$\text{(i)} \qquad A_{N-n+1}^{-1}(x_n - B_{N-n+1}y_n) = x_{n+1} \qquad 1 \leqq n \leqq N$$

$$\text{(ii)} \qquad y_n \in R^p \qquad 1 \leqq n \leqq N.$$

A simple linear algebraic calculation shows us that

$$T_{n\pi_n^*}(s_n) = [K_{n+1}^{-1} - (K_{n+1} + (B_n R_n^{-1} B_n')^{-1})^{-1}] K_{n+1} A_n s_n .$$

Thus the coefficient matrix is nonsingular. That is,

$$T_{n\pi_n^*} \in \mathcal{G}(S_n, S_{n+1}) \qquad 1 \leqq n \leqq N .$$

Therefore, the REVERSE THEOREM gives the reversed control process $\mathcal{D}_{-1}$ optimal reward functions $\{v^0, v^1, \cdots, v^N\}$ and an optimal policy $\{\hat{\sigma}_1, \hat{\sigma}_2, \cdots, \hat{\sigma}_N\}$, where

$$\begin{aligned} v^{N-n+1}(s_{N-n+2}) &= u^{n-1}(s_{N-n+2}) \\ &= (s_{N-n+2}, K_{N-n+2}s_{N-n+2}) \\ \hat{\sigma}_n(s_{N-n+2}) &= \pi_{N-n+1}^* \circ (T_{N-n+1, \pi_{N-n+1}^*})^{-1}(s_{N-n+1}) \\ &= -R_{N-n+1}^{-1} B_{N-n+1}' K_{N-n+2} s_{N-n+2} . \end{aligned}$$

Of course the optimal solutions are also obtained by solving the recursive formula:

$$\begin{aligned} & v^0(s_1) = (s_1, K_1 s_1) \\ & v^{N-n+1}(s_{N-n+2}) = \underset{a_{N-n+1} \in R^p}{\mathrm{Max}} [-(a_{N-n+1}, R_{N-n+1} a_{N-n+1}) \\ & \qquad -(A_{N-n+1}^{-1}(s_{N-n+2} - B_{N-n+1} a_{N-n+1}), \\ & \qquad Q_{N-n+1}(A_{N-n+1}^{-1}(s_{N-n+2} - B_{N-n+1} a_{N-n+1}))) \\ & \qquad + v^{N-n}(A_{N-n+1}^{-1}(s_{N-n+2} - B_{N-n+1} a_{N-n+1}))] \\ & \qquad s_{N-n+1} \in R^p, \qquad 1 \leqq n \leqq N . \end{aligned}$$

Concluding this section we remark that the 'reversible' result and analysis stated above also remain valid for many deterministic control processes [2, p. 329, p. 330], [3, p. 193, p. 195] and [4, p. 20, p. 21], which correspond to the case $K=0$ in our process.

5. A terminal control process

We consider the case in which $R_1 = R_2 = \cdots = R_{N+1} = R$ and each $f_n(s_n, a_n, s_{n+1})$ is the identity function on the interval R for the main DP $\mathcal{D}$ in § 2. We call this restriced DP the *terminal DP*. In particular we consider the following problem as a terminal DP (see [2, p. 305]): Let p and N be positive integers. Let b be a p-vector and A, B be two $p \times p$-nonsingular matrices with $AB \neq BA$.

Maximize

$$(Z_N Z_{N-1} \cdots Z_1 c,\ b)$$

(5.1) subject to

(i) $$Z_n = A \text{ or } B \qquad 1 \leqq n \leqq N \quad (c \in R^p).$$

This problem is represented by an N-stage main DP $\mathscr{D}=(\text{Max},\ \{S_n\}_1^{N+1},\ \{R_n\}_1^{N+1},\ \{A_n\}_1^N,\ \{f_n\}_1^N,\ k,\ \{T_n\}_1^N)$ starting at initial state $s_1=c$, where

$$S_n = R^p, \qquad R_n = R^1, \qquad A_n = \{A,\ B\},$$

$$f_n(s_n,\ a_n,\ s_{n+1})(r_{n+1}) = r_{n+1},$$

$$k(s_{N+1}) = (s_{N+1},\ b),$$

$$T_n(a_n)(s_n) = a_n s_n .$$

We call $\mathscr{D}$ the *main terminal control process.* The recursive formula becomes

$$u^{N-n+1}(s_n) = \text{Max}\,(u^{N-n}(As_n),\ u^{N-n}(Bs_n)) \qquad s_n \in R^p \quad 1 \leqq n \leqq N$$

$$u^0(s_{N+1}) = (s_{N+1},\ b) \qquad s_{N+1} \in R^p .$$

Thus the main terminal control process $\mathscr{D}$ has optimal reward functions $\{u^0,\ u^1,\ \cdots,\ u^N\}$ and an optimal policy $\{\pi_1^*,\ \pi_2^*,\ \cdots,\ \pi_N^*\}$, where $\pi_n^*(s_n)=A$ or B according as $u^{N-n}(As_n) \geqq u^{N-n}(Bs_n)$ or not. The desired maximum value of the problem (5.1) is $u^N(c)$.

On the hand, the problem

Minimize

$$u^N(Z_1^{-1} Z_2^{-1} \cdots Z_N^{-1} c)$$

(5.2) subjec to

(i) $$Z_n = A \text{ or } B \qquad N \geqq n \geqq 1 \quad (c \in R^p)$$

is represented by the reversed DP $\mathscr{D}_{-1}=(\text{Min},\ \{S_n\}_{N+1}^1,\ \{R_n\}_{N+1}^1,\ \{A_n\}_N^1,\ \{(f_n^{(s_n, a_n, s_{n+1})})^{-1}\}_N^1,\ u^N,\ \{(T_{na_n})^{-1}\}_N^1)$ starting at terminal (with respect to $\mathscr{D}$) state $s_{N+1}=c$, where

$$(f_n^{(s_n, a_n, s_{n+1})})^{-1}(r_n) = r_n$$

$$(T_{na_n})^{-1}(s_{n+1}) = a_n^{-1} s_{n+1} .$$

We call $\mathscr{D}_{-1}$ the *reversed terminal control process.* If $T_{n\pi_n^*} \in \mathscr{G}(S_n,\ S_{n+1})$, then the REVERSE THEOREM yields the reversed terminal control process $\mathscr{D}_{-1}$ optimal reward functions $\{u^N,\ u^{N-1},\ \cdots,\ u^0\}$ and an optimal policy $\{\pi_N^* \circ (T_{N\pi_N^*})^{-1},\ \pi_{N-1}^* \circ (T_{N-1,\pi_{N-1}^*})^{-1},\ \cdots,\ \pi_1^* \circ (T_{1\pi_1^*})^{-1}\}$. Therefore, the problem (5.2) has the minimum value $k(c)=(c,\ b)$. Note that the recursive formula becomes

$$v^{N-n+1}(s_{N-n+2}) = \text{Min}\,(v^{N-n}(A^{-1}s_{N-n+2}),\ v^{N-n}(B^{-1}s_{N-n+2}))$$

$$s_{N-n+2} \in R^p \qquad 1 \leqq n \leqq N$$

$$v^0(s_1) = u^N(s_1) \qquad s_1 \in R^p .$$

References

[1] R. BELLMAN, Dynamic Programming, Princeton University Press, Princeton, N. J., 1957.
[2] R. BELLMAN, Introduction to Matrix Analysis, McGraw-Hill, New York, 1960; 2nd ed., 1970.
[3] R. BELLMAN, Introduction to the Mathematical Theory of Control Processes, Vol. I, Linear Equations and Quadratic Criteria, Academic Press, New York, 1968.
[4] R. BELLMAN, Introduction to the Mathematical Theory of Control Process, Vol. II, Nonlinear Processes, Academic Press, New York, 1971.
[5] S. IWAMOTO, *Some operations on dynamic programming with one-dimensional state space*, J. Math. Anal. Appl., **69** (1979), 263-282.
[6] S. IWAMOTO, *Reversed dynamic programming on general state spaces*, Mem. Fac. Sci. Kyushu Univ. Ser. A, **32** (1978), 267-276.