Available online at www.sciencedirect.com
SCIENCE DIRECT
An International Journal
computers & mathematics with applications
Computers and Mathematics with Applications 48 (2004) 419–427
www.elsevier.com/locate/camwa

Dynamic Programming and Minimal Norm Solutions of Least Squares Problems

R. KALABA AND H. NATSUYAMA
Department of Biomedical Engineering
University of Southern California
Los Angeles, CA 90089, U.S.A.

(Received September 2003; accepted January 2004)

Keywords—Dynamic programming, Least squares, Pseudoinverses.

INTRODUCTION

Least squares problems occur widely in regression analysis, parameter estimation, analytical mechanics, and in many other areas. In [1], we introduced a new dynamic programming approach to least squares problems. The algorithm of that paper relied heavily on knowing the rank of the given matrix and knowing columns which are linearly independent. This paper extends the previous one by removing these restrictions. We develop a new algorithm which we call the $\alpha Q\beta R$ algorithm.

This formulation introduces *two* cost functions, which is new to dynamic programming literature. The first cost function is the square of the length of the current discrepancy vector, and the second is the square of the length of the current solution vector. The two cost functions are to be minimized simultaneously by optimally selecting the minimum length vector solution.

Finally, a connection with Greville's formula for generalized inverses is indicated.

PRINCIPLE OF OPTIMALITY

Let A be an $m \times n$ matrix, b be a column vector of dimension m, and x be a vector of dimension n. Given the matrix A and the vector b, we wish to determine the vector x such that $|Ax - b|^2$ is a minimum and the length of x is as small as possible. There are many approaches to this optimization problem [2]. Here we shall provide an approach through dynamic programming [3]. The reader may wish to consult [4–8].

We introduce *two* cost functions. First we write

$$f_k(b) = \text{the smallest square of the length of the residual vector } A_k x^k - b. \tag{1}$$

Here A_k is a matrix consisting of the first k columns of A and x^k is a column vector of dimension k. We also introduce

$$g_k(b) = \text{the smallest square of the length of the vector } x^k,$$

 Typeset by $\mathcal{A}_\mathcal{M}\mathcal{S}$-TeX
doi:10.1016/j.camwa.2004.01.007

where x^k is subject to the restriction

$$\left|A_k x^k - b\right|^2 = \min.\text{over } x^k. \tag{2}$$

In these definitions, $k = 1, 2, 3, \ldots, n$. We now obtain simultaneous recurrence relations for these functions. Having to use two cost functions is curious; yet, it seems unavoidable. We are led to new dynamic programming equations.

Suppose that $f_{k-1}(b)$ and $g_{k-1}(b)$ are known. We wish to obtain $f_k(b)$ and $g_k(b)$. We denote the individual columns of A by $a_1, a_2, \ldots, a_n$. There are two cases to consider, depending on whether a_k is linearly dependent on $a_1, a_2, \ldots, a_{k-1}$ or not. Assume first that a_k is linearly dependent on $a_1, a_2, \ldots, a_{k-1}$. In this case,

$$f_k(b) = f_{k-1}(b), \tag{3}$$

because a linear combination of $a_1, a_2, \ldots, a_{k-1}, a_k$ cannot be brought closer to b than a linear combination of $a_1, a_2, \ldots, a_{k-1}$. We also see that

$$g_k(b) =^{\min}_{x_k} \left[x_k^2 + g_{k-1}(b - x_k a_k)\right], \tag{4}$$

where x_k is a scalar. This follows because if x_k is the k^{th} component of x^k, then the term in square brackets is the square of x_k plus the smallest square of the length of a vector x^{k-1}, where $|A_{k-1}x^{k-1} - (b - x_k a_k)|^2 = \min.\text{over } x^{k-1}$.

Next we assume that a_k is linearly independent of the vectors $a_1, a_2, \ldots, a_{k-1}$. In this case, we must choose x_k, the k^{th} component of x^k in the approximation of b by $A_k x^k$, so that we minimize $f_{k-1}(b - x_k a_k)$. The reason is that with any choice of the scalar x_k, we must approximate the new target vector $b - x_k a_k$ as well as possible through choice of the sum $x_1 a_1 + \cdots + x_{k-1} a_{k-1}$. Thus, we may write

$$f_k(b) =^{\min}_{x_k} f_{k-1}(b - x_k a_k). \tag{5}$$

If the minimizing value of the scalar x_k is x_k^*, then we also have

$$g_k(b) = (x_k^*)^2 + g_{k-1}\left(b - x_k^* a_k\right). \tag{6}$$

Equations (3)–(6) constitute the desired system of recurrence relations. Equations (3) and (4) apply if a_k is dependent on $a_1, a_2, \ldots, a_{k-1}$, and equations (5) and (6) apply if a_k is independent of the earlier columns of the matrix A. The underlying role of Bellman's principle of optimality is clear [3].

In addition, for the case $k = 1$, we have

$$f_1(b) =^{\min}_{x_1} (a_1 x_1 - b)^\top (a_1 x_1 - b), \qquad a \neq 0. \tag{7}$$

Thus,

$$f_1(b) =^{\min}_{x_1} \left[a_1^\top a_1 x_1^2 - 2a_1^\top b x_1 + b^\top b\right]. \tag{8}$$

The minimizing condition is

$$a_1^\top a_1 x_1 - a_1^\top b = 0, \tag{9}$$

so that

$$x_1^* = \frac{a_1^\top b}{a_1^\top a_1} = a_1^+ b. \tag{10}$$

Here we have used the fact that the generalized inverse of the vector a_1, a_1^+, is $a_1^\top / a_1^\top a_1$ (assuming that $a_1 \neq 0$). It follows that

$$\begin{aligned} f_1(b) &= \left[b^\top \left(a_1^+\right)^\top a_1^\top a_1 a_1^+ b - 2b^\top a_1 a_1^+ b + b^\top b\right], \\ f_1(b) &= b^\top \left[I - a_1 a_1^+\right] b. \end{aligned} \tag{11}$$

For $g_1(b)$, we have

$$g_1(b) = b^\top \left(a_1^+\right)^\top a_1^+ b. \tag{12}$$

Thus, we see that $f_1(b)$ and $g_1(b)$ are quadratic forms in b which we may write as

$$f_1(b) = b^\top Q_1 b, \tag{13}$$

$$g_1(b) = b^\top R_1 b, \tag{14}$$

where Q_1 and R_1 are symmetric positive semidefinite $m \times m$ matrices.

RECURRENCE RELATIONS

We next show that the functions $f_k(b)$ and $g_k(b)$, $k = 1, 2, \ldots, n$, are positive semidefinite quadratic forms in b. As we have seen, this is true for $k = 1$. We complete the proof by induction by showing that if

$$f_{k-1}(b) = b^\top Q_{k-1} b \tag{15}$$

and

$$g_{k-1}(b) = b^\top R_{k-1} b, \tag{16}$$

then we also have

$$f_k(b) = b^\top Q_k b \tag{17}$$

and

$$g_k(b) = b^\top R_k b, \tag{18}$$

where the matrices Q_k and R_k are symmetric and positive semidefinite.

A. Dependent Vectors

First let us assume that the vector a_k is linearly dependent on the vectors $a_1, a_2, \ldots, a_{k-1}$. Then, by virtue of equation (3), we have

$$Q_k = Q_{k-1}. \tag{19}$$

From equation (4), we see that

$$\begin{aligned} g_k(b) &= \min_{x_k} \left[x_k^2 + (b - x_k a_k)^\top R_{k-1} (b - x_k a_k)\right] \\ &= \min_{x_k} \left[\left(1 + a_k^\top R_{k-1} a_k\right) x_k^2 + b^\top R_{k-1} b - 2 b^\top R_{k-1} a_k x_k\right]. \end{aligned} \tag{20}$$

From the first-order condition for minimization, we find

$$x_k = \frac{b^\top R_{k-1} a_k}{1 + a_k^\top R_{k-1} a_k}. \tag{21}$$

Upon substituting this value for x_k in equation (20), we find, after some simplification,

$$g_k(b) = b^\top R_k b, \tag{22}$$

where

$$R_k = R_{k-1} - \frac{R_{k-1} a_k a_k^\top R_{k-1}}{1 + a_k^\top R_{k-1} a_k}. \tag{23}$$

In equation (23), the denominator is never zero since R_{k-1} is assumed to be positive semidefinite. If we introduce the vector β_k by

$$\beta_k = R_{k-1} a_k, \tag{24}$$

then we may write

$$R_k = R_{k-1} - \frac{\beta_k \beta_k^\top}{1 + a_k^\top \beta_k}. \tag{25}$$

It is clear that the right side of the above equation,

$$R_{k-1} - \frac{\beta_k \beta_k^\top}{1 + a_k^\top \beta_k}, \tag{26}$$

is symmetric, and from the definition of $g_k(b)$ in equation (20) must be positive semidefinite. Equation (21) also takes the form

$$x_k = \frac{b^\top \beta_k}{1 + a_k^\top \beta_k}. \tag{27}$$

B. Independent Vectors

Now let us pass to the case in which a_k is linearly independent of the vectors $a_1, a_2, \dots, a_{k-1}$. Equations (5) and (6) now come into play. Also, a criterion for whether or not a_k is linearly dependent on the earlier vectors, $a_1, a_2, \dots, a_{k-1}$, will emerge. We see that

$$f_k(b) =_{x_k}^{\min} (b - x_k a_k)^\top Q_{k-1} (b - x_k a_k). \tag{28}$$

Since the expression on the right is merely a quadratic function of the scalar x_k, differentiation yields the condition for optimality that

$$x_k = \frac{b^\top Q_{k-1} a_k}{a_k^\top Q_{k-1} a_k}. \tag{29}$$

Later we shall see that, in this case, the denominator is actually positive. Substituting this value for x_k into equation (28) yields

$$f_k(b) = b^\top Q_k b, \tag{30}$$

where

$$Q_k = Q_{k-1} - \frac{Q_{k-1} a_k a_k^\top Q_{k-1}}{a_k^\top Q_{k-1} a_k}. \tag{31}$$

By introducing the vector

$$\alpha_k = Q_{k-1} a_k, \tag{32}$$

equation (31) takes the form

$$Q_k = Q_{k-1} - \frac{\alpha_k \alpha_k^\top}{\alpha_k^\top a_k}, \tag{33}$$

and equation (29) becomes

$$x_k = \frac{\alpha_k^\top b}{a_k^\top \alpha_k}. \tag{34}$$

We now wish to show, by induction, that α_k is the component of a_k that is orthogonal to the vectors $a_1, a_2, \dots, a_{k-1}$. We define $Q_0 = I$ and have

$$\alpha_1 = Q_0 a_1 = a_1,$$

$$Q_1 = Q_0 - \frac{\alpha_1 \alpha_1^\top}{a_1^\top \alpha_1}, \tag{35}$$

$$Q_1 = I - \frac{\alpha_1 \alpha_1^\top}{\alpha_1^\top \alpha_1}. \tag{36}$$

It follows that

$$Q_1 a_1 = 0 \tag{37}$$

and

$$Q_1 v = v, \qquad \left(v \text{ such that } \alpha_1^\top v = 0\right). \tag{38}$$

Thus, $Q_1 a_2 = \alpha_2$ is the component of a_2 that is orthogonal to $a_1 = \alpha_1$. Thus, a_2 has the form $a_2 = \alpha_2 + s a_1$, where s is a scalar.

To complete the inductive proof, we assume that α_l is the component of a_l that is orthogonal to the vectors $a_1, a_2, \ldots, a_{l-1}$, for $l = 2, 3, \ldots, k$. We must show that α_{k+1} is the component of a_{k+1} that is orthogonal to the vectors $a_1, a_2, \ldots, a_{k-1}, a_k$. By definition,

$$\begin{aligned} \alpha_{k+1} &= Q_k a_{k+1} \\ &= \left(Q_{k-1} - \frac{\alpha_k \alpha_k^\top}{a_k^\top \alpha_k}\right) a_{k+1}. \end{aligned} \tag{39}$$

First we see that

$$0 = \left(Q_{k-1} - \frac{\alpha_k \alpha_k^\top}{a_k^\top \alpha_k}\right) a_p, \tag{40}$$

where $p = 1, 2, 3, \ldots, k-1$. This is because α_k is orthogonal to $a_1, a_2, \ldots, a_{k-1}$. Furthermore, $Q_{k-1} a_p$ is the component of a_p which is orthogonal to the vectors $a_1, a_2, \ldots, a_{k-1}$. This component, of course, is the null vector. When $p = k$, the equality above holds because

$$Q_{k-1} a_k - \left(\frac{\alpha_k \alpha_k^\top}{a_k^\top \alpha_k}\right) a_k = \alpha_k - \alpha_k = 0. \tag{41}$$

Thus, α_{k+1} is orthogonal to the vectors $a_1, a_2, \ldots, a_k$. In addition,

$$Q_k = I - \sum_{l=1}^{k}{}' \frac{\alpha_l \alpha_l^\top}{a_l^\top \alpha_l}, \tag{42}$$

where the prime indicates that terms with zero denominators are omitted, so that if v is a vector such that $\alpha_l^\top v = 0$, $l = 1, 2, 3, \ldots, k$, then $Q_k v = v$. Thus,

$$Q_k \alpha_{k+1} = \alpha_{k+1}. \tag{43}$$

It follows that α_{k+1} is the component of a_{k+1} that is orthogonal to the vectors $a_1, a_2, \ldots, a_k$. Thus,

$$a_{k+1} = \alpha_{k+1} + \text{lin. comb. of } a_1, a_2, \ldots, a_k. \tag{44}$$

From the above representation, we also see that

$$\alpha_{k+1}^\top a_{k+1} = \alpha_{k+1}^\top \alpha_{k+1}, \qquad k = 0, 1, 2, \ldots, n-1. \tag{45}$$

Thus, the basic recurrence relation

$$Q_k = Q_{k-1} - \frac{\alpha_k \alpha_k^\top}{a_k^\top \alpha_k} \tag{46}$$

may be restated as

$$Q_k = Q_{k-1} - \frac{\alpha_k \alpha_k^\top}{\alpha_k^\top \alpha_k}, \qquad k = 1, 2, \ldots, n. \tag{46'}$$

From the discussion above, it is clear that the determination of whether or not the vector a_k is linearly dependent on the set of vectors $a_1, a_2, \ldots, a_{k-1}$ depends upon the vector α_k. If $\alpha_k = 0$, then a_k is linearly dependent upon the vectors $a_1, a_2, \dots, a_{k-1}$. Otherwise, it is independent of them.

The recurrence relation (6) leads to the recurrence relation for R_k,

$$R_k = \left(\alpha_k^+\right)^\top \alpha_k^+ + \left(I - a_k\alpha_k^+\right)^\top R_{k-1}\left(I - a_k\alpha_k^+\right). \tag{46''}$$

And the equation for x_k becomes

$$\begin{aligned} x_k &= \frac{\alpha_k^\top b}{\alpha_k^\top \alpha_k} \\ &= \alpha_k^+ b. \end{aligned} \tag{46'''}$$

THE $\alpha Q\beta R$ ALGORITHM

Let us now specify the $\alpha Q\beta R$ algorithm for solving the least squares problem $Ax \cong b$. There are two sweeps, one forward and one backward.

A. Forward Sweep to Compute and Store Auxiliary Vectors

In the forward sweep, we set

$$\alpha_1 = a_1, \tag{47}$$

$$Q_1 = I - \frac{\alpha_1\alpha_1^\top}{\alpha_1^\top\alpha_1} = I - a_1\alpha_1^+, \tag{48}$$

$$R_1 = \left(a_1^+\right)^\top a_1^+. \tag{49}$$

Then, for each value of k, $k = 2, 3, \dots, n$, there are two cases. If

$$\alpha_k = Q_{k-1}a_k = 0, \tag{50}$$

then

$$Q_k = Q_{k-1}, \tag{51}$$

$$\beta_k = R_{k-1}a_k, \tag{52}$$

$$R_k = R_{k-1} - \frac{\beta_k\beta_k^\top}{1 + a_k^\top\beta_k}. \tag{53}$$

If, on the other hand,

$$\alpha_k = Q_{k-1}a_k \neq 0, \tag{54}$$

then

$$\alpha_k^+ = \frac{\alpha_k^\top}{\alpha_k^\top\alpha_k}, \tag{55}$$

$$Q_k = Q_{k-1} - \alpha_k\alpha_k^+, \tag{56}$$

$$R_k = \left(\alpha_k^+\right)^\top \alpha_k^+ + \left(I - a_k\alpha_k^+\right)^\top R_{k-1}\left(I - a_k\alpha_k^+\right). \tag{57}$$

Only the vectors αs and βs need to be saved for the backward sweep.

B. Return Sweep to Compute Minimal Norm Vector Solution

In the return sweep, the components of the vector x, namely, $x_n, x_{n-1}, \dots, x_1$, are determined in that reverse order as follows. First we put

$$b_n = b. \tag{58}$$

Then, if $\alpha_n = 0$,

$$x_n = \frac{b_n^\top \beta_n}{1 + a_n^\top \beta_n}. \tag{59}$$

But if $\alpha_n \neq 0$, then

$$x_n = \frac{b_n^\top \alpha_n}{\alpha_n^\top \alpha_n}. \tag{60}$$

Next, for $k = n-1, n-2, \ldots, 1$, we set

$$b_k = b_{k+1} - x_{k+1} a_{k+1} \tag{61}$$

and

$$x_k = \frac{b_k^\top \beta_k}{\left(1 + a_k^\top \beta_k\right)}, \qquad \text{if } \alpha_k = 0 \tag{62}$$

or

$$x_k = \frac{b_k^\top \alpha_k}{\alpha_k^\top \alpha_k}, \qquad \text{if } \alpha_k \neq 0. \tag{63}$$

C. Procedure for $\alpha Q \beta R$ Dynamic Programming Algorithm

The algorithm is described by the following procedure.

1. Input the A matrix and the b vector
2. Sweep forward from columns 1 through n and store the $n\alpha$ and $n\beta$ vectors
    a. Column 1
        i. Initialize α_1, Q_1, and R_1 using equations (47)–(49)
        ii. Define Q_{k-1} and R_{k-1}
    b. Column $k = 2, 3, \ldots, n$
        i. Compute α_k using equation (32)
        ii. Test length of α_k against a tolerance
            (a) If length is less than tolerance, use equations (51)–(53) to compute Q_k, β_k, and R_k, and store
            (b) If length is greater than tolerance, use equations (55)–(57) to compute Q_k and R_k, and store; no β_k is needed
        iii. Shift current Q_k and R_k into Q_{k-1} and R_{k-1}
3. Sweep backward and determine the components of the vector x from the n^{th} component to the first
    a. Initialize the b_k vector for $k = n$
        i. Set $b_n = b$
        ii. Compute x_n using (59) or (60)
    b. Component $k = n-1, n-2, \ldots, 1$
        i. Modify b_k using equation (61)
        ii. Test length of α_k against tolerance
            (a) If length is less than tolerance, use equation (62) to compute x_k
            (b) If length is greater than tolerance, use equation (63) to compute x_k
4. Output x and other results as desired

GENERALIZED INVERSES AND $\alpha Q \beta R$

In view of the fact that the solution of the minimal norm least squares problem $(Ax-b)^\top(Ax-b) = \min$ can be obtained by the $aQbR$ algorithm, it is natural to seek the pseudoinverse of A, denoted A^+, through the algorithm. We now show how this may be done. The key to doing this is the Greville sequential method [8].

Greville's algorithm shows how to pass from a knowledge of A_{k-1}^+, the pseudoinverse of A_{k-1}, to the pseudoinverse A_k^+ of the matrix A_k. There are, of course, two cases to consider. In Greville's algorithm, the determination is made by considering the vector $c = (I - A_{k-1}A_{k-1}^+)a_k$. The vector c represents the component of a_k that is orthogonal to the columns of A_{k-1}. We see this from $a_k = c + A_{k-1}A_{k-1}^+ a_k$. That the vector c is orthogonal to the columns of A_{k-1} follows from

$$\begin{aligned} c^\top A_{k-1} &= a_k^\top \left(I - A_{k-1}A_{k-1}^+\right) A_{k-1} \\ &= 0. \end{aligned} \tag{64}$$

First let us consider the case in which $c \neq 0$. This means that the vector a_k has a component that is orthogonal to the vectors $a_1, a_2, \ldots, a_{k-1}$. Consequently, this is the case in which a_k is not linearly dependent upon $a_1, a_2, \ldots, a_{k-1}$.

The Greville updating is given by

$$A_k^+ = \begin{pmatrix} A_{k-1}^+ - A_{k-1}^+ a_k c^+ \\ c^+ \end{pmatrix}, \tag{65}$$

which requires a knowledge of the vector c, in addition to A_{k-1}^+ and a_k. But the $\alpha Q \beta R$ algorithm provides the vector α_k, which is the component of a_k that is orthogonal to $a_1, a_2, \ldots, a_{k-1}$, as we saw earlier. Thus, when A_{k-1}^+, a_k, and α_k are known, we may determine A_k^+ as

$$A_k^+ = \begin{pmatrix} A_{k-1}^+ - A_{k-1}^+ a_k \alpha_k^+ \\ \alpha_k^+ \end{pmatrix}. \tag{66}$$

The second case is that in which $c = 0$. In this case, $a_k = A_{k-1}A_{k-1}^+ a_k$, so that the vector a_k is linearly dependent on the columns of the matrix A_{k-1}. Greville's updating is given by the formula

$$A_k^+ = \begin{pmatrix} A_{k-1}^+ - A_{k-1}^+ a_k v^\top \\ v^\top \end{pmatrix}, \tag{67}$$

where

$$v = \frac{\left(A_{k-1}^+\right)^\top A_{k-1}^+ a_k}{1 + a_k^\top \left(A_{k-1}^+\right)^\top A_{k-1}^+ a_k}. \tag{68}$$

To interpret these relations from the point of view of the $\alpha Q \beta R$ algorithm, we recall that

$$\begin{aligned} g_k(b) &= \text{the smallest square of the length of the vector } x_k \text{ which minimizes } \left|A_k x^k - b\right|^2 \\ &= b^\top R_k b. \end{aligned} \tag{69}$$

On the other hand, we know the solution is

$$x^k = A_k^+ b, \tag{70}$$

so that we may write

$$g_k(b) = b^\top \left(A_k^+\right)^\top A_k^+ b. \tag{71}$$

Thus, we see that

$$R_k = \left(A_k^+\right)^\top A_k^+, \qquad k = 1, 2, \ldots, n. \tag{72}$$

This enables us to write

$$v = \frac{R_{k-1}a_k}{1 + a_k^\top R_{k-1} a_k}, \tag{73}$$

$$v = \frac{\beta_k}{1 + a_k^\top \beta_k}, \tag{74}$$

which expresses the vector v in terms of β_k.

These relations show how the updating of A_{k-1}^+ to A_k^+ is accomplished in terms of either α_k or β_k, depending on whether $\alpha_k \neq 0$ or $\alpha_k = 0$.

DISCUSSION

The new algorithm has been tested and used in a number of ways. Computational experience—with testing for dependence/independence, effect of near-dependency of vectors, accumulation of round-off errors, and applications in physical and estimation problems [4–8]—is being gained, and results are being reported in the literature. Clearly, much more remains to be done in this area.

The two cost functions introduced here are new to dynamic programming [3], and promise greater extensions of the theory in the future. We know the properties of the auxiliary matrices, the Qs, of symmetry, positive semidefiniteness, and idempotency. Yet, while this paper advances the theory of dynamic programming, it raises a number of questions. Are there interpretations for the βs and the Rs similar to those for the αs and Qs? Do the Qs and Rs *directly* give the generalized inverse, which plays a major role in least squares problems? Qs and Rs contain, clearly, information that is equivalent to that of the generalized inverse. We shall expand on this in a subsequent paper.

REFERENCES

1. R.E. Kalaba, H. Natsuyama, S. Ueno and R. Xu, The Bellman-Gauss principle for constrained motion, *Computers Math. Applic.* **37** (11/12), 1–7, (1999).
2. C. Lawson and R. Hanson, *Solving Least Squares Problems*, Prentice-Hall, Englewood Cliffs, NJ, (1974).
3. R. Bellman and R.E. Kalaba, *Dynamic Programming and Modern Control Theory*, Academic Press, New York, (1965).
4. R.E. Kalaba, H.H. Natsuyama and S. Ueno, The estimation of parameters in time-dependent transport problems: Dynamic programming and associative memories, *Computers Math. Applic.* **37** (2), 41–45, (1999).
5. R.E. Kalaba, H.H. Natsuyama and S. Ueno, Regression analysis via dynamic programming: I. Theory, In *The 30^{th} ISCIE International Symposium on Stochastic Systems Theory and Its Applications*, (November 1998).
6. R.E. Kalaba, H.H. Natsuyama and S. Ueno, Regression analysis via dynamic programming: II. Computational results, In *The 30^{th} ISCIE International Symposium on Stochastic Systems Theory and Its Applications*, (November 1998).
7. C. Itiki, R.E. Kalaba and H. Natsuyama, Estimation of parameters of a lower limb model, *Proc. Fourth Int. Symp. on Artificial Life, Computers Math. Applic.* **37**, 1–7, (1999).
8. F. Udwadia and R. Kalaba, *Analytical Dynamics*, Cambridge Univ. Press, New York, (1996).