IOSR Journal of Mathematics (IOSR-JM)
e-ISSN: 2278-5728, p-ISSN:2319-765X. Volume 10, Issue 3 Ver. IV (May-Jun. 2014), PP 16-22
www.iosrjournals.org

On The Extended Conjugate Gradient Method(ECGM) Algorithm For Discrete Optimal Control Problems And Some Of Its Features

F. M. Aderibigbe[*] And J.S. Apanapudor[**]
[*]*Department Of Mathematical Sciences, Ekiti State University, Ado-Ekiti*
[**]*Department Of Mathematics And Computer Science, Delta State University, Abraka*

Abstract: *The acceptability of an algorithm is a function of its implementability and convergence. In this paper, we examine some features of the extended conjugate gradient method (ECGM) algorithm, one of the optimization techniques for solving continuous/or discrete optimal control problems It is observed while using this algorithm, there is a consistent demand for some of the features of the algorithm. Among these are the step-size, alpha, the gradient(the partial derivatives), the search directions e.t.c. One of these features closely examined is the computation of* ∇J *, the gradient of* J *, the performance index* $J = \langle z, Hz \rangle, z = (x,u)^T$ *, which is foremost while implementing the algorithm. In the light of this, we develop an explicit expression for* ∇J *.Furthermore, a generalization of the expression for* ∇J *, for all positive integers n was attained, via mathematical induction.*
Keywords*: Step-size, Operator,Conjugate Search Directions*

I. Introduction

Most algorithms for solving discrete optimal control problems based on a class of descent methods, demand gradient evaluation of the performance functional. Efficient, within this class are Steepest descent method (SDM), Fletcher-Reeves method (FRM), Polak-Rebiere method (PRM), Newton methods and the Extended Conjugate gradient method (ECGM). However, none of these algorithms have been able to provide an explicit expression for ∇J , the gradient of the performance functional. It is in the light of this, that we desire to present an explicit expression for computing the partial derivatives ofperformance functional J ,where $J = \langle z, Hz \rangle, z = (x,u)^T$ .Let us consider the class of optimal control problems

Min $J_k(x,u) = \sum_{i=1}^{k} f(x_i,u_i), k < \infty$ (1.1)

with dynamic constraint of type $x_i = Ax_{i-1} + Bu_{i-1}$,(see Oliviera (2002), Polak (1971)).

This class of problems which fits into a closed loop or feedback control system and maintains an output level to a desired value without interference or fluctuation are known as *regulator problems*.Such problems often emanate from systems like water storage and supply engineering. In such systems, the state of the system at any instant automatically sets the control. This implies that the state is "fed back" to the control mechanism which adjusts itself without external influence, Ibiejugba (1985). Thus in solving these problems, we shall be interested in finding a control $\hat{u} \in \mathcal{R}^m$ and a corresponding trajectory $\hat{x} \in \mathcal{R}^n$, such that the cost functional

$$J_k(x,u) = \sum_{i=1}^{k} f(x_i,u_i) \quad (1.2)$$

is minimized over a class of all admissible control and state vectors, where $f: \mathcal{R}^n \times \mathcal{R}^m \to \mathcal{R}$ is continuously differentiable and k, denotes the duration of the control process. Let us consider a direct numerical solution to the linear quadratic optimal control problem formed as we let $f(x_i,u_i) = x_i^T P x_i + u_i^T Q u_i$ in equation (1.2) be subject to some discrete time linear dynamical constraint. Then, the resulting problem may be stated as

Minimize $J_k(x_i,u_i) = \sum_{i=1}^{k} [x_i^T P x_i + u_i^T Q u_i]$ (1.3)

Subject to $x_i = Ax_{i-1} + Du_{i-1}$ (1.4) where $x_i \in \mathcal{R}^n$, $u_i \in \mathcal{R}^m$ P and Q are nxn, mxm symmetric positive definite constant matrices respectively with A and D both constant matrices. The conventional penalty function method demands the transformation of the constrained problem in equations (1.3) and (1.4) into an unconstrained problem with the introduction of the penalty constant $\varphi > 0$,(Macki and Straues(1980)). Hence we have,

$$\text{Min } J_k(x_i, u_i) = \sum_{i=1}^{k} [x_i^T P x_i + u_i^T Q u_i + \varphi \langle x_i - Ax_{i-1} - Du_{i-1}, x_i - Ax_{i-1} - Du_{i-1} \rangle] \tag{1.5}$$

where $\varphi > 0$, the penalty constant, the superscript T denotes the transpose of a designated vector and the symbol $\langle .,. \rangle$, denotes the inner product in a suitable Hilbert space.

Now associate with equation (1.5) the control operator $\tilde{H}$ such that

$$\langle z, \tilde{H} z \rangle_w = \sum_{i=1}^{k} [x_i^T P x_i + u_i^T Q u_i + \varphi \langle x_i - Ax_{i-1} - Du_{i-1}, x_i - Ax_{i-1} - Du_{i-1} \rangle] \tag{1.6}$$

where w is a real Hilbert space and $z = (x_0, x_1, x_2, x_3, \cdots, x_k, u_0, u_1, u_2, u_3, \cdots, u_k)^T$ and $\tilde{H}$ is control operator constructed by Otunta(2003). The right hand side of equation (1.6) is a quadratic form with the associated block matrix $\tilde{H}$, of order (2k+2) given as

$$\tilde{H} = \begin{bmatrix} F & N \\ N^T & B \end{bmatrix} \tag{1.7}$$

where F,N and B are matrices whose entries are defined as follows:
F is a square matrix of order (k+1), with entries f_{ij} given by

$$f_{11} = \varphi A^T A, \; f_{ij} = -\varphi A, \text{ for all i, j such that } |i-j| = 1,$$
$$\underset{\substack{j\neq 1 \\ i\neq 1}}{f_{ij}} = p + \varphi(I + A^T A) \;, \; f_{k+1k+1} = p + \varphi I, \tag{1.8}$$

where I is an identity matrix of appropriate dimension with respect to that of A.
N is a square matrix of order (k+1) with entries defined as

$$\underset{j\neq k+1}{n_{ij}} = \varphi A^T D \text{, for all i, j such that } i = 1 + j, n_{ij} = 0, otherwise \tag{1.9}$$

N^T is the usual transpose of the matrix N.
B is a square diagonal matrix of order (k+1) with entries,

$$\underset{\substack{j\neq 1 \\ j\neq k+1}}{b_{ij}} = q + \varphi D^T D, b_{11} = \varphi D^T D, b_{k+1k+1} = q. \tag{1.10}$$

With this control operator $\tilde{H}$, we can conveniently solve our problem in equation (1.6).

The rest parts of the paper is outlined as follows: section two discusses the development of the explicit expression for ∇J ,section three examinesthe generalization of our expression for ∇J as required by the Extended Conjugate Gradient Method algorithm on Discrete Optimal Control Problems and we proceeds concluding comments in section four.

II. Development of An Explicit Expression for the gradient Computation

In solving equation (1.6), we will at various times, demand the evaluation of the derivative of $J = \langle z, Hz \rangle$. Thus it becomes pertinent to develop an explicit expression for its evaluation.. Knowing that every polynomial function f is differentiable and that every polynomial function f of degree greater than one is at least twice differentiable, (Taha(1996)), we proceed withour development by considering the one dimensional problem as below.

$$\text{Minimize } \sum_{i=1}^{k} [r x_i^2 + q u_i^2] \tag{2.1a}$$

$$\text{Subject to } x_i = v x_{i-1} + s u_{i-1} \tag{2.1b}$$

x_0 specified, where r,q,v and s are constants.

By the conventional penalty function method(Polak(1971)), the constrained problem in equation (2.1) is converted to the unconstrained problem

$$\text{Minimize } \sum_{i=1}^{k}[rx_i^2 + qu_i^2 + \varphi(x_i - vx_{i-1} - su_{i-1})^2] \tag{2.2}$$

where $\varphi > 0$ is the penalty constant.

By associating equation (2.2) with the quadratic functional $J = \langle z, Hz \rangle$,defined on the real Hilbert space w, we have,

$$J(x,u,\varphi) = \langle z, Hz \rangle_w = \sum_{i=1}^{k}[rx_i^2 + qu_i^2 + \varphi(x_i - vx_{i-1} - su_{i-1})^2] \tag{2.3}$$

On expanding equation (2.3), we have,

$$J_k(x,u,\varphi) = \langle z, Hz \rangle_w = \sum_{i=1}^{k}[rx_i^2 + qu_i^2 + \varphi(x_i^2 + v^2x_{i-1}^2 + s^2u_{i-1}^2 - 2vx_ix_{i-1} - 2sx_iu_{i-1} + 2vsx_{i-1}u_{i-1})]$$

(2.4) and to avoid any form of ambiguity, loss of purpose and contradiction in using J_k,we shall henceforth use J , in its place. Thus,

When $k = 1$, equation (2.4) becomes,

$$J(x,u,\varphi) = rx_1^2 + qu_1^2 + \varphi(x_1^2 + v^2x_0^2 + s^2u_0^2 - 2vx_1x_0 - 2sx_1u_0 + 2vsx_0u_0) \tag{2.5}$$

From equation (2.5), Let $J = J_1 + \varphi J_2$, (2.6)

where $J_1(x,u,\varphi) = rx_1^2 + qu_1^2$, (2.7)

$$J_2(x,u,\varphi) = x_1^2 + v^2x_0^2 + s^2u_0^2 - 2vx_1x_0 - 2sx_1u_0 + 2vsx_0u_0 \tag{2.8}$$

$$\text{Then } \nabla J(x,u,\varphi) = \left(\frac{\partial J_1}{\partial x_j} + \frac{\partial J_2}{\partial z_j}\right) + \varphi\left(\frac{\partial J_1}{\partial x_j} + \frac{\partial J_2}{\partial z_j}\right), j = 0,1 \tag{2.9}$$

So that

$$\nabla_{x_j} J = \left(\frac{\partial J_1}{\partial x_j} + \varphi\frac{\partial J_2}{\partial x_j}\right), j = 0,1.$$

(2.10)

$$\nabla_{u_j} J = \left(\frac{\partial J_1}{\partial u_j} + \varphi\frac{\partial J_2}{\partial u_j}\right), j = 0,1. \tag{2.11}$$

Thus using equations (2.7),(2.8), (2.10) and (2.11),we have,

$$\begin{aligned} \nabla_{x_0} J &= \frac{\partial J}{\partial x_0} = \varphi(2x_0v^2 - 2vx_1 + 2vsu_0) \\ \nabla_{x_1} J &= 2rx_1 + \varphi(2x_1 - 2vx_0 - 2su_0) \\ \nabla_{u_0} J &= \varphi(2u_0s^2 - 2sx_1 + 2vsx_0) \\ \nabla_{u_1} J &= 2qu_1 \end{aligned} \tag{2.12}$$

When $k = 2$, equation (2.4) become,

$$J = \langle z, Hz \rangle_w = \sum_{i=1}^{2}[rx_i^2 + qu_i^2 + \varphi(x_i^2 + v^2x_{i-1}^2 + s^2u_{i-1}^2 - 2vx_ix_{i-1} - 2sx_iu_{i-1} + 2vsx_{i-1}u_{i-1})]$$

$$J(x,u,\varphi) = \sum_{i=1}^{2}[rx_i^2 + qu_i^2 + \varphi(x_i^2 + v^2x_{i-1}^2 + s^2u_{i-1}^2 - 2vx_ix_{i-1} - 2sx_iu_{i-1} + 2vsx_{i-1}u_{i-1})] \tag{2.13}$$

$$= r(x_1^2 + x_2^2) + q(u_1^2 + u_2^2) + \varphi \begin{bmatrix} (x_1^2 + x_2^2) + v^2(x_0^2 + x_1^2) + s^2(u_0^2 + u_1^2) \\ -2vx_1(x_0 + x_2) - 2s(x_1u_0 + x_2u_1) + 2vs(x_0u_0 + x_1u_1) \end{bmatrix}$$

$$J_1(x,u,\varphi) = r(x_1^2 + x_2^2) + q(u_1^2 + u_2^2) \tag{2.14}$$

$$J_2(x,u,\varphi) = \begin{bmatrix} (x_1^2 + x_2^2) + v^2(x_0^2 + x_1^2) + s^2(u_0^2 + u_1^2) \\ -2vx_1(x_0 + x_2) - 2s(x_1u_0 + x_2u_1) + 2vs(x_0u_0 + x_1u_1) \end{bmatrix} \tag{2.15}$$

From equations (2.14) and (2.15), we have,

$$\nabla J(x,u,\varphi) = \sum_{j=0}^{2}\left[\left(\frac{\partial J_1}{\partial x_j} + \frac{\partial J_2}{\partial z_j}\right) + \varphi\left(\frac{\partial J_1}{\partial x_j} + \frac{\partial J_2}{\partial z_j}\right)\right], j = 0,1,2. \tag{2.16}$$

So that

$$\nabla_{x_j} J = \left(\frac{\partial J_1}{\partial x_j} + \varphi\frac{\partial J_2}{\partial x_j}\right), j = 0,1,2.$$

(2.17)

$$\nabla_{u_j} J = \left(\frac{\partial J_1}{\partial u_j} + \varphi\frac{\partial J_2}{\partial u_j}\right), j = 0,1,2. \tag{2.18}$$

Thus for $j = 0,1,2$

$$\nabla_{x_0} J = \frac{\partial J}{\partial x_0} = \varphi(2x_0v^2 - 2vx_1 + 2vsu_0) \ \nabla_{x_1} J = 2rx_1 + \varphi(2x_1(1+v^2) - 2v(x_0 + x_2) - 2su_0 + 2vsu_1)$$

$$\nabla_{x_2} J = \frac{\partial J}{\partial x_2} = 2rx_2 + \varphi(2x_2 - 2vx_1 - 2su_1) \ \nabla_{u_0} J = \frac{\partial J}{\partial u_0} = \varphi(2u_0s^2 - 2sx_1 + 2vsx_0)$$

$$\nabla_{u_1} J = \frac{\partial J}{\partial u_1} = 2qu_1 + \varphi(2u_1s^2 - 2sx_2 + 2vsx_1) \ \nabla_{u_2} J = \frac{\partial J}{\partial u_2} = 2qu_2$$

When $K = k$, equation (2.4) becomes,

$$J = \langle z, Hz\rangle_w = \sum_{i=1}^{k}[rx_i^2 + qu_i^2 + \varphi(x_i^2 + v^2x_{i-1}^2 + s^2u_{i-1}^2 - 2vx_ix_{i-1} - 2sx_iu_{i-1} + 2vsx_{i-1}u_{i-1})] \tag{2.19}$$

$$= r(x_1^2 + x_2^2 + \cdots + x_k^2) + q(u_1^2 + u_2^2 + \cdots + u_k^2) +$$

$$\varphi\begin{bmatrix} (x_1^2 + x_2^2 + \cdots + x_k^2) + v^2(x_0^2 + x_1^2 + \cdots + x_{k-1}^2) + s^2(u_0^2 + u_1^2 + u_2^2 + \cdots + u_{k-1}^2) - \\ 2v(x_1x_0 + \cdots + x_kx_{k-1}) - 2s(x_1u_0 + \cdots + x_ku_{k-1}) + 2vs(x_0u_0 + \cdots + x_{k-1}u_{k-1}) \end{bmatrix}$$

$$= J_1 + \varphi J_2$$

where $J_1 = r(x_1^2 + x_2^2 + \cdots + x_k^2) + q(u_1^2 + u_2^2 + \cdots + u_k^2)$ (2.20)

$$J_2 = \varphi\begin{bmatrix} (x_1^2 + x_2^2 + \cdots + x_k^2) + v^2(x_0^2 + x_1^2 + \cdots + x_{k-1}^2) + s^2(u_0^2 + u_1^2 + u_2^2 + \cdots + u_{k-1}^2) - \\ 2v(x_1x_0 + \cdots + x_kx_{k-1}) - 2s(x_1u_0 + \cdots + x_ku_{k-1}) + 2vs(x_0u_0 + \cdots + x_{k-1}u_{k-1}) \end{bmatrix} \tag{2.21}$$

From equations (2.20) and (2.21), we have

$$\nabla J(x,u,\varphi) = \sum_{j=0}^{k}\left[\left(\frac{\partial J_1}{\partial x_j} + \frac{\partial J_2}{\partial z_j}\right) + \varphi\left(\frac{\partial J_1}{\partial x_j} + \frac{\partial J_2}{\partial z_j}\right)\right], j = 0,1,2,\cdots,k. \tag{2.22}$$

So

$$\nabla_{x_j} J = \left(\frac{\partial J_1}{\partial x_j} + \varphi\frac{\partial J_2}{\partial x_j}\right), j = 0,1,2,\cdots,k. \tag{2.23}$$

$$\nabla_{u_j} J = \left(\frac{\partial J_1}{\partial u_j} + \varphi\frac{\partial J_2}{\partial u_j}\right), j = 0,1,2,\cdots,k. \tag{2.24}$$

Using equations (2.23) and (2.24), for $j = 0,1,2,\cdots,k.$, we have,

$$\nabla_{x_0} J = \frac{\partial J}{\partial x_0} = \varphi(2x_0v^2 - 2vx_1 + 2vsu_0)\,, \nabla_{x_1} J = 2rx_1 + \varphi(2x_1(1+v^2) - 2v(x_0 + x_2) - 2su_0 + 2vsu_1)$$

$$\nabla_{x_2} J = \frac{\partial J}{\partial x_2} = 2rx_2 + \varphi(2(1+v^2)x_2 - 2vx_1 - 2su_1 + 2vsu_2)\,, \ldots$$

$$\nabla_{x_k} J = \frac{\partial J}{\partial x_k} = 2rx_k + \varphi(2(1+v^2)x_k - 2vx_{k-1} - 2su_{k-1} + 2vsu_{k-1})$$

$$\nabla_{u_0} J = \frac{\partial J}{\partial u_0} = \varphi(2u_0s^2 - 2sx_1 + 2vsx_0)\,, \nabla_{u_1} J = \varphi(2u_1s^2 - 2sx_2 + 2vsx_1)\,,\ \nabla_{u_k} J = \frac{\partial J}{\partial u_k} = 2qu_k$$

Therefore the expression below

$$\nabla J(x,u,\varphi) = \sum_{j=0}^{k}\left[\left(\frac{\partial J_1}{\partial x_j} + \frac{\partial J_2}{\partial z_j}\right) + \varphi\left(\frac{\partial J_1}{\partial x_j} + \frac{\partial J_2}{\partial z_j}\right)\right], j = 0,1,2,\cdots,k. \qquad (2.25)$$

is the explicit expression for generating the gradient of the cost functional (2.4).

III. Generalization Of The Explicit Expression For ∇J .

We present inthis section, the generalization of the expression in equation(2.11) using the idea of mathematical induction (Griffel(1993)). This is presented in the following theorem.

Theorem 3.1

If x_j and u_j are the state and control variables of a system; $\nabla_x J$ and $\nabla_u J$ are the respective partial derivatives of J with respect to x_j and u_j. Then, for $j = 0,1,2,\cdots,k-1.$, k is the duration of the control process, we have

$$\nabla J(x,u,\varphi) = \sum_{j=0}^{k}\left[\left(\frac{\partial J_1}{\partial x_j} + \frac{\partial J_2}{\partial z_j}\right) + \varphi\left(\frac{\partial J_1}{\partial x_j} + \frac{\partial J_2}{\partial z_j}\right)\right], j = 0,1,2,\cdots,k. \qquad (2.26)$$

is true.

Proof: We present the proof of this theorem using mathematical induction. Thus given, $\nabla J(x,u,\varphi) = \langle z, Hz\rangle$ as in equation (2.4) and using mathematical induction we establish the proof of the theorem as follows:

$$J = \langle z, Hz\rangle_w = \sum_{i=1}^{k}[rx_i^2 + qu_i^2 + \varphi(x_i^2 + v^2x_{i-1}^2 + s^2u_{i-1}^2 - 2vx_ix_{i-1} - 2sx_iu_{i-1} + 2vsx_{i-1}u_{i-1})]\,, (2.27)$$

Step 1: When $n = K = 1$, we have,

$$\nabla J(x,u,\varphi) = \sum_{j=0}^{1}\left[\left(\frac{\partial J_1}{\partial x_j} + \frac{\partial J_2}{\partial z_j}\right) + \varphi\left(\frac{\partial J_1}{\partial x_j} + \frac{\partial J_2}{\partial z_j}\right)\right], j = 0,1.$$

and with $J = J_1 + \varphi J_2$, where from equation (2.27), $J_1(x,u,\varphi) = rx_1^2 + qu_1^2$

$J_2(x,u,\varphi) = x_1^2 + v^2x_0^2 + s^2u_0^2 - 2vx_1x_0 - 2sx_1u_0 + 2vsx_0u_0$, we can obtain the gradient of J with respect to x and u as $\nabla_{x_j} J = \left(\frac{\partial J_1}{\partial x_j} + \varphi\frac{\partial J_2}{\partial x_j}\right)$ and $\nabla_{u_j} J = \left(\frac{\partial J_1}{\partial u_j} + \varphi\frac{\partial J_2}{\partial u_j}\right), j = 0,1.$ (2.28)

Step 2. Since it is true for $n = K = 1$, we assume next that it is true for $n = K = k$, i.e.

$$\nabla J(x,u,\varphi) = \sum_{j=0}^{k}\left[\left(\frac{\partial J_1}{\partial x_j} + \frac{\partial J_2}{\partial z_j}\right) + \varphi\left(\frac{\partial J_1}{\partial x_j} + \frac{\partial J_2}{\partial z_j}\right)\right], j = 0,1,2,\cdots,k.$$

Also $J = J_1 + \varphi J_2$, where $J_1 = r(x_1^2 + x_2^2 + \cdots + x_k^2) + q(u_1^2 + u_2^2 + \cdots + u_k^2)$

$$J_2 = \varphi \begin{bmatrix} (x_1^2 + x_2^2 + \cdots + x_k^2) + v^2(x_0^2 + x_1^2 + \cdots + x_{k-1}^2) + s^2(u_0^2 + u_1^2 + u_2^2 + \cdots + u_{k-1}^2) - \\ 2v(x_1x_0 + \cdots + x_kx_{k-1}) - 2s(x_1u_0 + \cdots + x_ku_{k-1}) + 2vs(x_0u_0 + \cdots + x_{k-1}u_{k-1}) \end{bmatrix}$$

From these we can obtain $\nabla_{x_j} J = \left(\frac{\partial J_1}{\partial x_j} + \varphi \frac{\partial J_2}{\partial x_j} \right)$ and $\nabla_{u_j} J = \left(\frac{\partial J_1}{\partial u_j} + \varphi \frac{\partial J_2}{\partial u_j} \right)$, for $j = 0,1,\cdots,k$.

Step 3. Next we show that it is true for $n = K = k+1$.

$$\nabla J(x,u,\varphi) = \sum_{j=0}^{1} \left[\left(\frac{\partial J_1}{\partial x_j} + \frac{\partial J_2}{\partial z_j} \right) + \varphi \left(\frac{\partial J_1}{\partial x_j} + \frac{\partial J_2}{\partial z_j} \right) \right] + \sum_{j=0}^{2} \left[\left(\frac{\partial J_1}{\partial x_j} + \frac{\partial J_2}{\partial z_j} \right) + \varphi \left(\frac{\partial J_1}{\partial x_j} + \frac{\partial J_2}{\partial z_j} \right) \right] +$$

$$\cdots + \sum_{j=0}^{k} \left[\left(\frac{\partial J_1}{\partial x_j} + \frac{\partial J_2}{\partial z_j} \right) + \varphi \left(\frac{\partial J_1}{\partial x_j} + \frac{\partial J_2}{\partial z_j} \right) \right] + \sum_{j=0}^{k+1} \left[\left(\frac{\partial J_1}{\partial x_j} + \frac{\partial J_2}{\partial z_j} \right) + \varphi \left(\frac{\partial J_1}{\partial x_j} + \frac{\partial J_2}{\partial z_j} \right) \right]$$

$$= \sum_{j=0}^{k+1} \left[\left(\frac{\partial J_1}{\partial x_j} + \frac{\partial J_2}{\partial z_j} \right) + \varphi \left(\frac{\partial J_1}{\partial x_j} + \frac{\partial J_2}{\partial z_j} \right) \right], \; j = 0,1,\cdots,k+1 \qquad (2.30)$$

andwith $J_1 = r(x_1^2 + x_2^2 + \cdots + x_{k+1}^2) + q(u_1^2 + u_2^2 + \cdots + u_{k+1}^2)$

$$J_2 = \varphi \begin{bmatrix} (x_1^2 + x_2^2 + \cdots + x_{k+1}^2) + v^2(x_0^2 + x_1^2 + \cdots + x_k^2) + s^2(u_0^2 + u_1^2 + u_2^2 + \cdots + u_k^2) - \\ 2v(x_1x_0 + \cdots + x_kx_{k+1}) - 2s(x_1u_0 + \cdots + x_{k+1}u_k) + 2vs(x_0u_0 + \cdots + x_ku_k) \end{bmatrix}$$

We can generate $\nabla_{x_j} J = \left(\frac{\partial J_1}{\partial x_j} + \varphi \frac{\partial J_2}{\partial x_j} \right)$ and $\nabla_{u_j} J = \left(\frac{\partial J_1}{\partial u_j} + \varphi \frac{\partial J_2}{\partial u_j} \right)$, for $j = 0,1,\cdots,k+1.$ (2.31)

Since the above expression is true for $n = K = k+1$, we conclude that it is true for all integers n.
This completes the proof of the theorem above.

IV. Conclusion

An efficient explicit expression is proposed to enable us obtain the partial derivatives of $J(x,u,\varphi) = \langle z, Hz \rangle$, necessary for the application of the ECGM algorithm on DOCP. The main contributions of this paper are the development of the explicit expression and its generalization using mathematical induction.

References

[1]. Griffel, D.H.(1973). Applied functional Analysis, Kills Horwood Limited Chichester.
[2]. Ibiejugba, M.A.(1980) Computing Methods in Optimal Control, Ph. D Thesis University of Leeds, England.
[3]. Ibiejugba, M.A. (1985) Computing Gradient descent of a Smooth functional, Advances in Modeling and Simulation, Vol.4.pp. 37-43.
[4]. Macki, J. and Straus, A.(1980) Introuction to Optimal Control Theory, Springer-Verlag, N.Y. Inc
[5]. Oliviera, I.B.(2002) A "HUM" Conjugate Gradient Algorithm for Constrained Non-linear
[6]. Optimal Control: Terminal and Regulator Problems, APh.D Thesis presented to the Institute of Technology, Massahusetts.
[7]. Onwuatu,J.E.(2000) Optimal Control of discrete systems with delays, Journal of Nigeria Mathematical Society, Vol.19,pp. 39 – 57.
[8]. Otunta, F.O.(2003) A Gradient Method for Discrete Optimal Control Problems, Abacus , J. of MAN. Vol. 3., N0.2
[9]. Otunta, F.O.(1991) Optimization Techniques for a class of Regulator Problems, PhD Thesis, University of Ilorin, Nigeria
[10]. Polak,E. (1971) Computational Methods in Optimization Academic Press London.
[11]. Taha, H.(1996) Operations Research,