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 etc. One of these features closely examined is the computation of $\nabla J$, 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 $\nabla J$. Furthermore, a generalization of the expression for $\nabla J$, 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 (SD), 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 $\nabla J$, 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 of performance functional $J$, where $J = \langle z, Hz \rangle$, $z = (x, u)^T$. Let us consider the class of optimal control problems

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

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

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

is minimized over a class of all admissible control and state vectors, where $f: \mathcal{R}^n \times \mathcal{R}^m \rightarrow \mathcal{R}$ 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

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

Subject to $x_i = Ax_{i-1} + Du_{i-1}$

References

where \( x_i \in \mathbb{R}^n \), \( u_i \in \mathbb{R}^m \) 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 Straus(1980)). Hence we have,

\[
\text{Min } J_k(x_i, u_i) = \sum_{i=1}^{k} \left[ x_i^T P x_i + u_i^T Q u_i + \varphi (x_i - Ax_{i-1} - Du_{i-1}, x_i - Ax_{i-1} - Du_{i-1}) \right]
\]  

(1.5)

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

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

\[
\langle z, \tilde{H}z \rangle_w = \sum_{i=1}^{k} \left[ x_i^T P x_i + u_i^T Q u_i + \varphi (x_i - Ax_{i-1} - Du_{i-1}, x_i - Ax_{i-1} - Du_{i-1}) \right]
\]  

(1.6)

where \( w \) is a real Hilbert space and \( z = (x_0, x_1, x_2, \cdots, x_k, u_0, u_1, u_2, \cdots, u_k)^T \) 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

\[
\tilde{H} = \begin{bmatrix} F & N \\ N^T & B \end{bmatrix}
\]  

(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
  \[
  f_{i1} = \varphi A^T A, \quad f_{ij} = -\varphi A, \text{ for all } i, j \text{ such that } |i - j| = 1.
  \]

- \( N \) is a square matrix of order \((k+1)\) with entries defined as
  \[
  n_{ij} = \varphi A^T D, \quad \text{for all } i, j \text{ such that } i = 1 + j, n_{ij} = 0, \text{otherwise}
  \]

(1.9)

- \( N^T \) is the usual transpose of the matrix \( N \).

- \( B \) is a square diagonal matrix of order \((k+1)\) with entries,
  \[
  b_{ij} = q + \varphi D^T D, \quad b_{ii} = \varphi D^T D, \quad b_{i+k+1} = q.
  \]

(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 \( VJ \) section three examines the generalization of our expression for \( VJ \) 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 without development by considering the one dimensional problem as below.

\[
\text{Minimize } \sum_{i=1}^{k} [rx_i^2 + qu_i^2]
\]  

(2.1a)

Subject to \( x_i = v x_{i-1} + su_{i-1} \)

(2.1b)
On The Extended Conjugate Gradient Method (ECGM) Algorithm For Discrete Optimal …

Let \( x_0 \) be 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

\[
\text{Minimize } \sum_{i=1}^{k} \left[ r^2_i + q^2_i + \phi(x_i - vx_{i-1} - su_{i-1})^2 \right]
\]  
(2.2)

where \( \phi > 0 \) 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

\[
J(x, u, \phi) = \langle z, Hz \rangle_w = \sum_{i=1}^{k} \left[ r^2_i + q^2_i + \phi(x_i - vx_{i-1} - su_{i-1})^2 \right]
\]  
(2.3)

On expanding equation (2.3), we have,

\[
J_k(x, u, \phi) = \langle z, Hz \rangle_w = \sum_{i=1}^{k} \left[ r^2_i + q^2_i + \phi(x_i^2 + v^2x_{i-1}^2 + s^2u_{i-1}^2 - 2vx_ix_{i-1} - 2sx_{i-1} + 2vxs_{i-1}u_{i-1}) \right]
\]  
(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,

\[
J(x, u, \phi) = r^2_1 + q^2_1 + \phi(x_1^2 + v^2x_0^2 + s^2u_0^2 - 2vx_1x_0 - 2sx_0 + 2vxs_0u_0)
\]  
(2.5)

From equation (2.5), let \( J = J_1 + \phi J_2 \),

where

\[
J_1(x, u, \phi) = r^2_1 + q^2_1
\]  
(2.6)

\[
J_2(x, u, \phi) = x_1^2 + v^2x_0^2 + s^2u_0^2 - 2vx_1x_0 - 2sx_0 + 2vxs_0u_0
\]  
(2.7)

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

(2.8)

So that

\[
\nabla_{x_j} J = \left( \frac{\partial J_1}{\partial x_j} + \phi \frac{\partial J_2}{\partial z_j} \right), \quad j = 0, 1
\]  
(2.10)

\[
\nabla_{u_j} J = \left( \frac{\partial J_1}{\partial u_j} + \phi \frac{\partial J_2}{\partial z_j} \right), \quad j = 0, 1
\]  
(2.11)

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

\[
\nabla_{x_0} J = \phi(2vx_1^2 - 2vx_0 + 2vxs_0u_0)
\]

\[
\nabla_{x_1} J = 2rx_1 + \phi(2x_1 - 2vx_0 - 2sx_0)
\]  
(2.12)

\[
\nabla_{u_0} J = \phi(2u_0s^2 - 2sx_1 + 2vxs_0)
\]

\[
\nabla_{u_1} J = 2qu_1
\]

When \( k = 2 \), equation (2.4) becomes,

\[
J = \langle z, Hz \rangle_w = \sum_{i=1}^{2} \left[ r^2_i + q^2_i + \phi(x_i^2 + v^2x_{i-1}^2 + s^2u_{i-1}^2 - 2vx_ix_{i-1} - 2sx_{i-1} + 2vxs_{i-1}u_{i-1}) \right]
\]

\[
J(x, u, \phi) = \sum_{i=1}^{2} \left[ r^2_i + q^2_i + \phi(x_i^2 + v^2x_{i-1}^2 + s^2u_{i-1}^2 - 2vx_ix_{i-1} - 2sx_{i-1} + 2vxs_{i-1}u_{i-1}) \right]
\]  
(2.13)
\[ = r(x_1^2 + x_2^2) + q(u_1^2 + u_2^2) + \varphi \left[ (x_1^2 + x_2^2) + v^2 (x_0^2 + x_1^2) + s^2 (u_0^2 + u_1^2) 
- 2ux_1 (x_0 + x_2) - 2s(x_1 u_0 + x_2 u_1) + 2vs (x_0 u_0 + x_1 u_1) \right] \]
From equations (2.14) and (2.15), we have,

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

(2.16)

So that

\[ \nabla_{x_j} J = \left( \frac{\partial J_1}{\partial x_j} + \phi \frac{\partial J_2}{\partial z_j} \right), \quad j = 0,1,2. \]  

(2.17)

\[ \nabla_{u_j} J = \left( \frac{\partial J_1}{\partial u_j} + \phi \frac{\partial J_2}{\partial u_j} \right), \quad j = 0,1,2. \]  

(2.18)

Thus for \( j = 0,1,2 \)

\[ \nabla_{x_0} J = \frac{\partial J}{\partial x_0} = \phi(2x_0v^2 - 2vx_0 + 2vsu_0) \quad \nabla_{x_1} J = 2rx_1 + \phi(2x_1(1 + v^2) - 2(vx_0 + x_2) - 2su_0 + 2su_1) \]

\[ \nabla_{x_2} J = \frac{\partial J}{\partial x_2} = 2rx_2 + \phi(2x_2 - 2vx_0 - 2su_0) \quad \nabla_{u_0} J = \frac{\partial J}{\partial u_0} = \phi(2u_0s^2 - 2sx_1 + 2vsv_0) \]

\[ \nabla_{u_1} J = \frac{\partial J}{\partial u_1} = 2qu_1 + \phi(2u_1s - 2sx_2 + 2vsv_1) \quad \nabla_{u_2} J = \frac{\partial J}{\partial u_2} = 2qu_2 \]

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

\[ J = \langle z, Hz \rangle_w = \sum_{i=1}^{i=1} \left[ rx_i^2 + qu_i^2 + \phi(x_i^2 + v^2x_i + s^2u_i^2 - 2vx_ix_i - 2sx_iu_i + 2vx_iu_{i+1} - 2sv_x_iu_{i+1} + 2vx_iu_{i-1} + 2vsv_x_iu_{i-1} - 2sv_x_iu_{i+1}) \right] \]  

(2.19)

\[ = r(x_1^2 + x_2^2 + \cdots + x_k^2) + q(u_1^2 + u_2^2 + \cdots + u_k^2) + \phi \left[ (x_1^2 + x_2^2 + \cdots + x_k^2) + v^2(x_0^2 + x_1^2 + \cdots + x_{k-1}^2) + s^2(u_0^2 + u_1^2 + u_2^2 + \cdots + u_{k-1}^2) - 2vx_0x_1 + \cdots + 2vx_{k-1}x_1 + 2vx_0u_1 + \cdots + 2vx_{k-1}u_1 + 2vx_0u_{k-1} + \cdots + 2vx_{k-1}u_{k-1} \right] \]  

(2.20)

\[ J_1 = f_{1,1} + \phi f_{1,2} \]

\[ J_2 = f_{2,1} + \phi f_{2,2} \]

(2.21)

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

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

(2.22)

Thus,

\[ \nabla_{x_j} J = \left( \frac{\partial J_1}{\partial x_j} + \phi \frac{\partial J_2}{\partial z_j} \right), \quad j = 0,1,2,\cdots, k. \]  

(2.23)

\[ \nabla_{u_j} J = \left( \frac{\partial J_1}{\partial u_j} + \phi \frac{\partial J_2}{\partial u_j} \right), \quad j = 0,1,2,\cdots, k. \]  

(2.24)

Using equations (2.23) and (2.24), for \( j = 0,1,2,\cdots, k \), we have,
\[ \nabla_x J = \frac{\partial J}{\partial x_0} = \varphi(2x_0v^2 - 2vx_1 + 2vsu_0), \quad \nabla_y J = 2rx_1 + \varphi(2x_1(1 + v^2) - 2v(x_0 + x_2) - 2su_0 + 2vsu_t) \]
\[ \nabla_x J = \frac{\partial J}{\partial x_2} = 2rx_2 + \varphi(2(1 + v^2)x_2 - 2vx_1 - 2su_2 + 2vsu_x), \ldots \]
\[ \nabla_x J = \frac{\partial J}{\partial x_k} = 2rx_k + \varphi(2(1 + v^2)x_k - 2vx_{k-1} - 2su_{k-1} + 2vsu_{k-1}) \]
\[ \nabla_u J = \frac{\partial J}{\partial u_0} = \varphi(2u_0s^2 - 2sx_1 + 2vsx_0), \quad \nabla_y J = \varphi(2u_1s^2 - 2sx_2 + 2vsx_1), \quad \nabla_z J = \frac{\partial J}{\partial u_k} = 2qu_k \]

Therefore the expression below
\[ \nabla J(x, u, \varphi) = \sum_{j=0}^{k} \left[ \frac{\partial J_1}{\partial x_j} + \frac{\partial J_2}{\partial z_j} \right] + \varphi \left( \frac{\partial J_1}{\partial x_j} + \frac{\partial J_2}{\partial z_j} \right), \quad j = 0, 1, 2, \ldots, k. \] (2.25)

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

III. Generalization Of The Explicit Expression For \( \nabla J \).

We present in this 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 \( x \) and \( u \) 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 \( x \) and \( u \). Then, for \( j = 0, 1, 2, \ldots, k - 1 \), \( k \) is the duration of the control process, we have
\[ \nabla J(x, u, \varphi) = \sum_{j=0}^{k} \left[ \frac{\partial J_1}{\partial x_j} + \frac{\partial J_2}{\partial z_j} \right] + \varphi \left( \frac{\partial J_1}{\partial x_j} + \frac{\partial J_2}{\partial z_j} \right), \quad j = 0, 1, 2, \ldots, k. \] (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:

\[ J = \langle z, Hz \rangle = \sum_{i=1}^{n} \langle r^2x_i + qu_i^2 + \varphi(x_i^2 + v^2x_{i-1} + s^2u_i^2 - 2vx_i x_{i-1} - 2sx_i u_{i-1} + 2vsx_{i-1}u_{i-1}) \rangle. \] (2.27)

Step 1: When \( n = K = 1 \), we have,
\[ \nabla J(x, u, \varphi) = \sum_{j=0}^{1} \left[ \frac{\partial J_1}{\partial x_j} + \frac{\partial J_2}{\partial z_j} \right] + \varphi \left( \frac{\partial J_1}{\partial x_j} + \frac{\partial J_2}{\partial z_j} \right), \quad j = 0, 1. \]

and with \( J = J_1 + \varphi J_2 \), where from equation (2.27), \( J_1(x, u, \varphi) = r^2x_i + qu_i^2 \)

\[ J_2(x, u, \varphi) = x_i^2 + v^2x_{i-1} + s^2u_i^2 - 2vx_i x_{i-1} - 2sx_i u_{i-1} + 2vsx_{i-1}u_{i-1} \]

we can obtain the gradient of \( J \) with respect to \( x \) and \( u \) as \( \nabla_x J = \left( \frac{\partial J_1}{\partial x_j} + \varphi \frac{\partial J_2}{\partial x_j} \right) \) and \( \nabla_u J = \left( \frac{\partial J_1}{\partial u_j} + \varphi \frac{\partial J_2}{\partial u_j} \right), \quad 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.
\[ \nabla J(x, u, \varphi) = \sum_{j=0}^{k} \left[ \frac{\partial J_1}{\partial x_j} + \frac{\partial J_2}{\partial z_j} \right] + \varphi \left( \frac{\partial J_1}{\partial x_j} + \frac{\partial J_2}{\partial z_j} \right), \quad j = 0, 1, 2, \ldots, k. \]

Also \( J = J_1 + \varphi J_2 \), where \( J_1 = r(x_i^2 + x_{i-1}^2 + \cdots + x_0^2) + q(u_i^2 + u_{i-1}^2 + \cdots + u_0^2) \)

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

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

and with

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

\[
J_2 = \varphi \left[ (x_1^2 + x_2^2 + \cdots + x_{k+1}^2) + v^2(x_0^2 + x_1^2 + \cdots + x_k^2) + s^2(u_0^2 + u_1^2 + u_2^2 + \cdots + u_k^2) - 2v(x_1x_0 + \cdots + x_{k+1}x_k) - 2s(x_0u_0 + \cdots + x_{k+1}u_k) + 2vs(x_0u_0 + \cdots + x_{k+1}u_k) \right]
\]

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

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 \( J(x,u, \varphi) = (z, Hz) \), 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