Computing Techniques For The Conjugate Search Directions Of The Ecgm Algorithm For Optimal Control Problems

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

Abstract: In this paper, we examine techniques for the construction of the conjugate search direction, \( p_n \) and \( p_w \), often required in the implementation of the Extended Conjugate Gradient Method (ECGM) for optimal control problems. The various techniques were derived analytically using some ideas from numerical linear algebra. We also establish the authenticity of these approaches by presenting a proof via mathematical induction, which when applied for the computation of these vectors proved successful most especially for the Discrete Optimal Control Problems (DOCP).

Keywords: Conjugate Search Direction, Operator, Parameter,

I. Introduction

Optimal Control Problems entails the finding of a control vector \( u \) and a corresponding state \( x \), which enhances the minimization or the maximization of the performance index or criteria. One class of methods that cannot be omitted when discussing the solution of optimal control problems is the conjugate gradient methods. Conjugate gradient methods represent an important class of unconstrained optimization algorithms, with special attention to global convergence properties (Hager and Zhang (2006)). This family of algorithms includes a lot of variants, well known in the literature, with important convergence properties and numerical efficiency. One of these variants is the Extended Conjugate Gradient Method (ECGM) algorithm, proposed by Ibiejugba and Onumanyi (1984), based on the formalism of the conjugate gradient method (CGM) algorithm due to Hestenes and Stiefel (1952). Since then several authors have worked on the algorithm with the intent of improving the performance of the method. To mention but a few, Aderibigbe, (1988), dwelt on the implementation of the algorithm giving it a numerical behavior that has made others have confidence in it. An extension of the ECGM to control problems governed by linear differential delay equations was also presented (see Aderibigbe, (1995)). Otunta (1991) examined the convergence of the ECGM algorithm for continuous optimal control problems and in the process constructed a new control operator in line with Ibiejugba (1980). He also established a different control operator that enabled him use the ECGM algorithm to solve Discrete Optimal control Problems (DOCP). Olorunsola, (1992) further worked on discrete optimal control problems as one of the means of solving continuous optimal control problems. Olotu, (2010) worked on discretizing the constrained continuous optimal control problems. It is our desire to examine the construction and computation of the search directions in the ECGM algorithm with the intent of improving the performance of the method.

To accomplish this, consider the class of optimal control (regulator) problems of the form

\[
\text{Min} \ J(x,u) = \sum_{i=1}^{k} \left[ x_i^T P x_i + u_i^T Q u_i \right]
\]

Subject to

\[
x_i = A x_{i-1} + D u_{i-1}
\]

where \( x_i \in \mathbb{R}^n \), \( u_i \in \mathbb{R}^m \), \( P \) and \( Q \) are \( nxn \), \( mxm \) symmetric positive definite constant matrices respectively with \( A \) and \( D \) constant matrices. By the conventional penalty function method equation (1.1) is converted to an unconstrained problem with the introduction of the penalty constant \( \varphi > 0 \), as follows:

\[
\text{Min} \ J(x,u) = \sum_{i=1}^{k} \left[ x_i^T P x_i + u_i^T Q u_i + \varphi \left( x_i - A x_{i-1} - D u_{i-1}, x_i - A x_{i-1} - D u_{i-1} \right) \right]
\]

In order to solve the unconstrained problem, we associate with equation (1.2) the quadratic functional \( \langle z, Hz \rangle \) to have,

\[
\langle z, Hz \rangle = J(x,u, \varphi) = \sum_{i=1}^{k} \left[ x_i^T P x_i + u_i^T Q u_i + \varphi (x_i - A x_{i-1} - D u_{i-1})^2 \right]
\]
where $W$ is a real Hilbert space, $z = (x_0, x_1, x_2, \ldots, x_k, u_0, u_1, u_2, \ldots, u_k)^T$. $H$ is a control operator derived by Otunta (2003). The right hand side of equation (1.3) is a quadratic form with the associated block matrix $H$, of order $(k+1)$ given as follows

$$H = \begin{pmatrix} F & N \\ N^T & B \end{pmatrix} \quad (1.4)$$

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

$$f_{ij} = \varphi A^T A, f_{ij} = -\varphi A, \text{for all } i,j \text{ such that } |i-j|=1,$$

$$f_{ij} = P + \varphi(I + A^T A), f_{ij} = 0, \text{ otherwise}, f_{k+1,k+1} = P + \varphi I, \quad (1.5)$$

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

$$n_{ij} = \varphi A^T D, n_{ij} = -\varphi D, \text{ for all } i,j \text{ such that } i=1+j, n_{ij} = 0, \text{ otherwise} \quad (1.6)$$

$N^T$ is the usual transpose of the matrix $N$.

$B$ is a square diagonal matrix of order $k+1$ with entries,

$$b_{ij} = Q + \varphi D^T D, b_{11} = \varphi D^T D, b_{k+1,k+1} = Q. \quad (1.7)$$

With this control operator $H$ (see Otunta (2003)), we can now implement the ECGM algorithm to solve the problem on equation (1.3)

The outline of the rest part of our paper is as follows. In Section 2, we begin with a recall of the Extended Conjugate Gradient Method Algorithm, and the necessary tools for the construction of conjugate search directions. Also consider some theorems that shed some light on the ways of constructing the search directions. In section 3, we present the techniques for constructing the $H$. In section 4, we look at the implementation procedure of the ECGM algorithm on a one-dimensional discrete optimal control problem. Section 5, present our conclusion and indicate areas of future research.

II. Necessary Tools For The Construction Of $p_x$ And $p_u$ In The ECGM Algorithm

The ECGM algorithm as proposed by Ibiejugba and Onumayi (1984) for solving the equation (1.3) is as follows:

Step 1: Initialize the sequence by guessing the first element $z(0) = (x(0), u(0))^T \in W$

Step 2: Set

$$\begin{cases} p_{x,0} = -g_{x,0} \\ p_{u,0} = -g_{u,0} \end{cases} \quad (2.1)$$

Step 3: Compute $x_{i+1} = x_i + \alpha_i p_{x,i}$ and $u_{i+1} = u_i + \alpha_i p_{u,i}$

$$\begin{cases} g_{x,i+1} = g_{x,i} + \alpha_i Ap_{x,i} \\ g_{u,i+1} = g_{u,i} + \alpha_i Ap_{u,i} \end{cases} \quad (2.2)$$

Next upgrade the gradient and descent directions by computing

$$\begin{cases} p_{x,i+1} = -g_{x,i+1} + \beta_{ix} p_{x} \\ p_{u,i+1} = -g_{u,i+1} + \beta_{iu} p_{u} \end{cases} \quad (2.3)$$

where $\alpha_i = \frac{\|g_i\|^2}{\langle p_i, Ap_i \rangle} = \frac{\langle g_i, g_i \rangle}{\langle p_i, Ap_i \rangle} \quad (2.4)$
Computing Techniques For The Conjugate Search Directions Of The Ecgm Algorithm For ...

\[ \beta_i = \frac{\|g_{i+1}\|^2}{\|g_i\|^2} = \frac{\langle g_{i+1}, g_{i+1} \rangle}{\langle g_i, g_i \rangle} \]  
\[ g_i = \begin{pmatrix} g_{x_i} \\ g_{u_i} \end{pmatrix}, \quad p_i = \begin{pmatrix} p_{x_i} \\ p_{u_i} \end{pmatrix} \]  
(2.8)

Step 4: If \( p_{x_i} \) or \( p_{u_i} = 0 \), or \( i = k \), the specified duration of control process. Go To step 5. Otherwise set \( i = i + 1 \), and Go To step 2.

Step 5: Stop and set \( z^* = (x^*_i, u^*_i)^T \)

Next we consider some basic concepts that will enable us undertake the construction of the gradient vectors and descent directions.

a. Conjugacy

Consider a quadratic function given by

\[ F(x) = \frac{1}{2} x^T G x + b^T x + c \]  
(2.10)

where \( G \) is a positive definite symmetric matrix, \( b \) a vector and \( c \) a scalar. Then the directions represented by two vectors \( u \neq 0 \) and \( v \neq 0 \) are conjugate (or orthogonal) with respect to \( G \) if

\[ u^T G v = 0 \]  
(2.11)

Geometrically (for simplicity within two-dimensions), the level curves

\[ F(x) = \psi \]  
(2.12)

for different values of \( \psi \) are concentric ellipses. The concept of conjugacy has its origin in the theory of poles and polars of an ellipse Hastenes (1980). The following theorems formulated on the properties of conjugate directions will be useful and so they are in order.

Theorem 1

If the vectors \( p_i \) are mutually conjugate (i.e. \( p_i^T G p_j = 0 \) for \( i \neq j \), for all \( i \) and \( j \)), then they are linearly independent.

This theorem implies that there exists at least one set of \( n \) independent vectors mutually conjugate with respect to the matrix \( G \); the set of eigenvectors of \( G \) forms such a set.

On the minimization of the function \( F(x) \) subject to \( x \in \mathbb{R}^n \), we state the theorem below without proof.

Theorem 2 (Navon and Legler (1987))

Suppose \( x_k \) and \( x_{k+1} \) are consecutive current points in a minimization of \( F(x) \). If

(i) \( x_k \) minimizes \( F(x) \) in direction \( p_l \).

(ii) \( x_{k+1} \) minimizes \( F(x) \) in the direction \( p_m \).

(iii) \( p_l \) and \( p_m \) are conjugate directions, then \( x_{k+1} \) also minimizes \( F(x) \) in the direction \( p_l \).

The above theorem most especially conditions (i) and (iii) implies that \( \{ p_k \} \), \( g_k = \nabla F(x_k) \) is the gradient of \( F(x_k) \) for \( i = 0, 1, \ldots, l \) and \( p_i^T G p_m = 0 \). Hence for a quadratic function \( F \), we have

\[ g_{k+1} - g_k = G(x_{k+1} - x_k) \]

and from (ii) \( x_{k+1} = x_k + \alpha_k p_k \)

where \( \alpha_k \) is determined by the line minimization

\[ F(x_k + \alpha_k p_k) = \min F(x_k + \alpha_k p_k) . \]

(b) Construction of a set of mutually conjugate directions

Given a set of linearly independent vectors \( u_0, u_1, \ldots, u_{n-1} \), we can construct a set of mutually \( G \)-conjugate directions \( p_0, p_1, \ldots, p_{n-1} \) by the following procedure. Set
Computing Techniques For The Conjugate Search Directions Of The ECGM Algorithm For ....

\[ p_0 = u_0 \]  
(2.13)

and then for \( i = 1, 2, \ldots, n - 1 \) successively define

\[ p_i = u_i + \sum_{j=0}^{i-1} a_{ij} p_j \]  
(2.14)

where \( a_{ij} \) are the coefficients chosen so that \( p_i \) is \( G \)-conjugate to the previous direction \( p_{i-1}, p_{i-2}, \cdots, p_0 \). This is possible if for \( l = 0, 1, 2, \cdots, i - 1 \),

\[ p_i^T G p_i = u_i^T G p_i + \sum_{j=0}^{i-1} a_{ij} p_j^T G p_i = 0 \]  
(2.15)

If previous coefficients \( a_{ij} \) were chosen so that at \( p_0, p_1, \cdots, p_{i-1} \) are \( G \)-conjugate, then we have

\[ g_j^T p_i = 0, \text{if } j \neq l \text{ and from (2.15) we have,} \]

\[ a_{ij} = -\frac{u_i^T G p_j}{p_j^T G p_j}, \text{for all } i = 1, 2, \ldots, n - 1, j = 1, 2, \cdots, i - 1. \]  
(2.16)

Hence the set of search directions \( p_0, p_1, \cdots, p_{n-1} \) defined by (2.13) - (2.16) is \( G \)-conjugate and the subspaces spanned by \( p_0, p_1, \cdots, p_i \) and \( u_0, u_1, \cdots, u_i \) are the same.

(c) For a better numerical performance of the ECGM for DOCP, we introduce the use of the Polak-Ribiere-Polyak (PRP), (1969) formula for updating the conjugate search directions \( \beta_k^{PRP} \), defined as follows:

\[ \beta_k^{PRP} = \frac{g_k^T (g_k - g_{k-1})}{\| g_{k-1} \|^2} \]  
(2.17)

Thus (2.17) will enable us to construct the conjugate search directions as presented in the following section.

III. Construction Of The Search Directions

In this section, we present the various ways of constructing or generating \( \{ p_k \} \), using some ideas of equations (2.13) to (2.17).

(I) One way of generating the conjugate search directions \( p_k \) is to use a linear combination of the current (negative) gradient direction and the previous search directions to produce a new search direction which is conjugate to all previous ones. Thus with the values \( p_{s_0} = -g_{s_0} \), \( p_{u_0} = -g_{u_0} \) as given in the algorithm, we can generate all other subsequent values of \( p_k \) by using

\[ p_k = -g_k + \beta_k^{PRP} p_{k-1} \]  
(3.1)

where

\[ \beta_k^{PRP} = \frac{g_k^T (g_k - g_{k-1})}{\| g_{k-1} \|^2} \text{ Polak(1971)} \]  
(3.2)

The iterates are given by \( z_{k+1} = z_k + \alpha_k p_k \), and \( g_k = \left( g(x_k), g(u_k) \right)^T \). Hence equation (3.1) becomes

\[ p_k = -g_k + \frac{g_k^T (g_k - g_{k-1})}{\| g_{k-1} \|^2} p_{k-1} \]  
(3.3)

where \( \| \| \) denotes the norm of vector in a suitable, Hilbert space \( W \), Griffel(1993) and so with \( p_k = (p_{x,k}, p_{u,k})^T \) we obtain
Computing Techniques For The Conjugate Search Directions Of The ECGM Algorithm For …

\[ p_{s,k} = -g_{s,k} + \beta_{s,k} p_{s,k-1}, \beta_{s,k} = \frac{g_{s,k}^T (g_{s,k} - g_{s,k-1})}{\|g_{s,k}\|^2} \]  

(3.4)

\[ p_{u,k} = -g_{u,k} + \beta_{u,k} p_{u,k-1}, \beta_{u,k} = \frac{g_{u,k}^T (g_{u,k} - g_{u,k-1})}{\|g_{u,k}\|^2} \]

(II) Another way of generating the conjugate search directions is presented in the theorem below.

**Theorem3.**

If \( g_i \) is the ith gradient of the sequence of gradients generated from equation (2.10), \( p_i \) ith the search direction in the sequence of conjugate search directions and assume that \( p_0 = -g_0 \) and the Polak-Ribiere-Polyak(PRPR)(1969) formula in equation(2.17). Then the descent search direction at the kth step is given by

\[ p_k = -\left\langle g_k, g_k \right\rangle \sum_{i=0}^{k} \frac{g_i}{\left\langle g_i, g_i \right\rangle} \]  

(3.5)

Proof: We present the proof of this theorem using mathematical induction. Let \( p_i \) be as defined previously in (2.18), \( p_0 = -g_0 \) and \( \left\langle g_i, g_j \right\rangle = 0 \) for all \( i \neq j \), then when \( i = 1 \),

\[ p_1 = -g_1 + \beta_1 p_0 = -g_1 + \frac{g_1^T (g_1 - g_0)}{\left\langle g_0, g_0 \right\rangle} p_0 \]

\[ = -g_1 + \left\langle g_1, g_1 \right\rangle p_0 = -g_1 + \left\langle g_1, g_1 \right\rangle - \left\langle g_1, g_0 \right\rangle \left\langle g_0, g_0 \right\rangle p_0 = -g_1 + \left\langle g_1, g_1 \right\rangle p_0 \]

\[ = -\left\langle g_1, g_1 \right\rangle g_1 - \left\langle g_1, g_1 \right\rangle g_0 = -\left\langle g_1, g_1 \right\rangle \left[ \frac{g_1}{\left\langle g_1, g_1 \right\rangle} + \frac{g_0}{\left\langle g_0, g_0 \right\rangle} \right] \]

\[ p_1 = -\left\langle g_1, g_1 \right\rangle \sum_{i=0}^{1} \frac{g_i}{\left\langle g_i, g_i \right\rangle} \]

when \( i = 2 \), \( p_2 = -g_2 + \beta_2 p_1 = -g_2 + \frac{g_2^T g_2}{\left\langle g_1, g_1 \right\rangle} p_1 = -\left\langle g_2, g_2 \right\rangle \sum_{i=0}^{2} \frac{g_i}{\left\langle g_i, g_i \right\rangle} \).

Assume that the above result is true for \( i = k \), i.e. \( p_1, \cdots, p_k \).

\[ P(k) = p_1 + p_2 + \cdots + p_k = \sum_{i=0}^{k} \frac{g_i}{\left\langle g_i, g_i \right\rangle} \]

(3.6)

Then we shall show that it is true for \( i = k + 1 \),

\[ P(k+1) = P_{k+1} = P_k + p_{k+1} = \sum_{i=0}^{k+1} \frac{g_i}{\left\langle g_i, g_i \right\rangle} \]

\[ = -g_{k+1} + \frac{g_{k+1}^T g_{k+1}}{\left\langle g_k, g_k \right\rangle} p_k \]

\[ = -g_{k+1} + \frac{g_{k+1}^T g_{k+1}}{\left\langle g_k, g_k \right\rangle} \left( -g_k + \frac{g_k^T g_k}{\left\langle g_{k-1}, g_{k-1} \right\rangle} p_{k-1} \right) \]

\[ = -g_{k+1} - \left\langle g_{k+1}, g_{k+1} \right\rangle g_k + \frac{g_{k+1}^T g_{k+1}}{\left\langle g_k, g_k \right\rangle} \frac{g_k^T g_k}{\left\langle g_{k-1}, g_{k-1} \right\rangle} p_{k-1} \]

www.iosrjournals.org 27 | Page
Computing Techniques For The Conjugate Search Directions Of The ECGM Algorithm For DOCP

\[
\begin{align*}
\frac{\langle g_{k+1} \rangle \cdot g_{k+1} \rangle}{\langle g_{k+1} \rangle \cdot g_{k+1} \rangle} g_{k+1} &= \frac{\langle g_{k+1} \rangle \cdot g_{k+1} \rangle}{\langle g_{k+1} \rangle \cdot g_{k+1} \rangle} g_{k+1} + \frac{\langle g_{k+1} \rangle \cdot g_{k+1} \rangle}{\langle g_{k+1} \rangle \cdot g_{k+1} \rangle} g_{k-2} + \frac{\langle g_{k+1} \rangle \cdot g_{k+1} \rangle}{\langle g_{k+1} \rangle \cdot g_{k+1} \rangle} p_{k-2} \\
&= -\frac{\langle g_{k+1} \rangle \cdot g_{k+1} \rangle}{\langle g_{k+1} \rangle \cdot g_{k+1} \rangle} g_{k+1} - \frac{\langle g_{k+1} \rangle \cdot g_{k+1} \rangle}{\langle g_{k+1} \rangle \cdot g_{k+1} \rangle} g_{k-1} + \frac{\langle g_{k+1} \rangle \cdot g_{k+1} \rangle}{\langle g_{k+1} \rangle \cdot g_{k+1} \rangle} g_{k-2} - \frac{\langle g_{k+1} \rangle \cdot g_{k+1} \rangle}{\langle g_{k+1} \rangle \cdot g_{k+1} \rangle} g_{k-3} - \\
&\cdots - \frac{\langle g_{k+1} \rangle \cdot g_{k+1} \rangle}{\langle g_{k+1} \rangle \cdot g_{k+1} \rangle} g_{k-j+1} - \frac{\langle g_{k+1} \rangle \cdot g_{k+1} \rangle}{\langle g_{k+1} \rangle \cdot g_{k+1} \rangle} g_{2} - \frac{\langle g_{k+1} \rangle \cdot g_{k+1} \rangle}{\langle g_{k+1} \rangle \cdot g_{k+1} \rangle} g_{1} - \frac{\langle g_{k+1} \rangle \cdot g_{k+1} \rangle}{\langle g_{k+1} \rangle \cdot g_{k+1} \rangle} g_{0} \\
&= -\frac{\langle g_{k+1} \rangle \cdot g_{k+1} \rangle}{\langle g_{k+1} \rangle \cdot g_{k+1} \rangle} \left[ \frac{g_{k}}{\langle g_{k+1} \rangle \cdot g_{k+1} \rangle} + \frac{g_{k-1}}{\langle g_{k+1} \rangle \cdot g_{k+1} \rangle} + \frac{g_{k-2}}{\langle g_{k+1} \rangle \cdot g_{k+1} \rangle} + \frac{g_{k-3}}{\langle g_{k+1} \rangle \cdot g_{k+1} \rangle} + \cdots \right] \\
&= -\frac{\langle g_{k+1} \rangle \cdot g_{k+1} \rangle}{\langle g_{k+1} \rangle \cdot g_{k+1} \rangle} \left[ \frac{\sum_{j=0}^{k} \frac{g_{j}}{\langle g_{k+1} \rangle \cdot g_{k+1} \rangle}}{\langle g_{k+1} \rangle \cdot g_{k+1} \rangle} \right] \\
&= -\langle g_{k+1} \rangle \cdot g_{k+1} \rangle \left[ \frac{\sum_{j=0}^{k} \frac{g_{j}}{\langle g_{k+1} \rangle \cdot g_{k+1} \rangle}}{\langle g_{k+1} \rangle \cdot g_{k+1} \rangle} \right] \quad (3.7)
\end{align*}
\]

As required is the (k+1)st step descent search direction for generating \( p_i = (p_j, p_n)^T \). Hence we have shown that the expression is true for all integers \( n \).

IV. Implementation Of The ECGM Algorithm For DOCP

We will now employ the foregoing in the computation of the search directions of the ECGM for discrete optimal control problems (DOCP). First we recall the algorithm proposed by Otunta (2003).

**DOCP Algorithm (Otunta 2003)**

Step 1: Select \( z_0 = (x_0, u_0)^T \) from \( w \), the domain of the problem, \( x_0 \) specified.

Step 2. Compute the partial derivatives of \( J \), in equation (3) with respect to \( x_i \) and \( u_j \).

Respectively for all \( i \). Set \( t = 0 \) and let \( p_{x_0} = -g_{x_0}, p_{u_0} = -g_{u_0} \)

Step 3. Compute \( \alpha_{x_i} = \frac{\langle g_{x_i}, g_{x_i} \rangle}{\langle p_{x_i}, H p_{x_i} \rangle}, \alpha_{u_i} = \frac{\langle g_{u_i}, g_{u_i} \rangle}{\langle p_{u_i}, H p_{u_i} \rangle} \)

with \( H g_{x_i} = H p_{x_i} \big|_{p_{x_i}=0} \) and \( H g_{u_i} = H p_{u_i} \big|_{p_{u_i}=0} \) \hspace{1cm} (4.1)

\[ x_{i+1} = x_i + \alpha_{x_i} p_{x_i}, u_{i+1} = u_i + \alpha_{u_i} p_{u_i} \]

and next...
Computing Techniques For The Conjugate Search Directions Of The ECGM Algorithm For ....

\[ P_{u,1} = -\left( g_{j,1}, g_{j,1} \right) \sum_{j=0}^{i+1} \frac{g_{s,j}}{g_{s,j}, g_{s,j}} \]

\[ P_{u,1} = -\left( g_{j,1}, g_{j,1} \right) \sum_{j=0}^{i+1} \frac{g_{u,j}}{g_{u,j}, g_{u,j}} \]  \hspace{1cm} (4.2)

Step 4If \( p_{u,1} \) or \( p_{u} = 0 \) or \( i = k \), the specified duration of the control process; GO TO step 5. Otherwise set \( i = i + 1 \) and Go To step 2.

Step 5 Stop and set \( z^* = (x^*_i, u^*_i) \)

Next consider a one-dimensional problem to illustrate how to apply the ECGM algorithm on DOCP.

Example I

One dimensional case

\[ \text{Minimize} \sum_{i=1}^{k} \left[ r x_i^2 + q u_i^2 \right] \]

Subject to

\[ x_i = \nu x_{i-1} + s u_{i-1}, x_0 \text{ specified,} \]

where \( r, q, \nu \) and \( s \) are constants

By the conventional penalty method, Ibiejugba, et al(1992), the constrained problem is converted to the unconstrained problem

\[ \text{Minimize} \sum_{i=1}^{k} \left[ r x_i^2 + q u_i^2 + \phi \left( x_i - \nu x_{i-1} - s u_{i-1} \right)^2 \right] \]  \hspace{1cm} (4.4)

Next associate with the quadratic functional \( J = \{ z, Hz \} \), defined on the real Hilbert space \( W \) in line with Ibiejugba(1980); this satisfies

\[ J = \{ z, Hz \} = \sum_{i=1}^{k} \left[ r x_i^2 + q u_i^2 + \phi \left( x_i - \nu x_{i-1} - s u_{i-1} \right)^2 \right] \]  \hspace{1cm} (4.5)

where \( w \) is a real Hilbert space Griffel(1993) and \( z = (x_0, x_1, x_2, \ldots, x_k, u_0, u_1, u_2, \ldots, u_k) \)

In order for us to make the application of this algorithm worthwhile, we require the partial derivatives of the quadratic functional and the elements of the matrix operator \( H \). Thus from equation (4.5) we obtain

\[ \nabla x_j J = \left( \frac{\partial J_1}{\partial x_j} + \phi \frac{\partial J_2}{\partial x_j} \right), \quad j = 0, 1, 2, \ldots, k. \] \hspace{1cm} (4.6a)

\[ \nabla u_j J = \left( \frac{\partial J_1}{\partial u_j} + \phi \frac{\partial J_2}{\partial u_j} \right), \quad j = 0, 1, 2, \ldots, k. \] \hspace{1cm} (4.6b)

Of importance also, are the elements of the matrix operator \( H \) defined in equations (1.5) – (1.7) which can now be obtained with respect to the one – dimensional case problem as follows:

\[ F = \begin{pmatrix}
\varphi v^2 & -\varphi v & 0 & 0 & \cdots & 0 & 0 \\
-\varphi v & r + \varphi (1 + v) & -\varphi v & 0 & \cdots & 0 & 0 \\
0 & -\varphi v & r + \varphi (1 + v) & -\varphi v & \cdots & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & 0 & \cdots & -\varphi v & r + \varphi
\end{pmatrix} \] \hspace{1cm} (4.8)
Computing Techniques For The Conjugate Search Directions Of The Ecgm Algorithm For ....

\[ N = \begin{pmatrix} \varphi vs & 0 & 0 & 0 & \cdots & 0 & 0 \\ -\varphi s & \varphi vs & 0 & 0 & \cdots & 0 & 0 \\ 0 & -\varphi s & \varphi vs & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & -\varphi s & 0 \end{pmatrix} \]

\[ B = \begin{pmatrix} \varphi s^2 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & q + \varphi s^2 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & q + \varphi s^2 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 0 & q \end{pmatrix} \]

Thus the matrix operator \( H \) takes the form:

\[ H = \begin{pmatrix} \varphi v^2 & -\varphi v & 0 & 0 & \cdots & 0 & 0 & \varphi vs & 0 & 0 & 0 & \cdots & 0 & 0 \\ -\varphi v & M & -\varphi v & 0 & \cdots & 0 & 0 & -\varphi s & \varphi vs & 0 & 0 & \cdots & 0 & 0 \\ 0 & -\varphi v & M & -\varphi v & \cdots & 0 & 0 & -\varphi s & \varphi vs & 0 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & -\varphi v & r + \varphi & 0 & 0 & 0 & \cdots & -\varphi s & 0 \end{pmatrix} \]

Where \( M = r + \varphi (1 + \nu), N = q + \varphi s^2 \). Then the matrix-vector product \( H p_i \)

\[ H p_i = \begin{pmatrix} \varphi v^2 & -\varphi v & 0 & 0 & \cdots & 0 & 0 & \varphi vs & 0 & 0 & 0 & \cdots & 0 & 0 \\ -\varphi v & M & -\varphi v & 0 & \cdots & 0 & 0 & -\varphi s & \varphi vs & 0 & 0 & \cdots & 0 & 0 \\ 0 & -\varphi v & M & -\varphi v & \cdots & 0 & 0 & -\varphi s & \varphi vs & 0 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & -\varphi v & r + \varphi & 0 & 0 & 0 & \cdots & -\varphi s & 0 \end{pmatrix} \begin{pmatrix} p_{i0} \\ p_{i1} \\ p_{i2} \\ \vdots \\ p_{iN} \\ p_{i0} \\ p_{i1} \\ p_{i2} \\ \vdots \\ p_{iN} \end{pmatrix}, \]

yields
Computing Techniques For The Conjugate Search Directions Of The Ecgm Algorithm For ...

\[ H_p = \begin{pmatrix}
\phi v^2 p_{x_0} - \phi v p_{n_1} + \phi v s p_{u_0} \\
-\phi v p_{x_0} + [r + \phi (1 + v)] p_{n_1} - \phi s p_{n_0} + \phi v s p_{u_1} \\
-\phi v p_{x_1} + [r + \phi (1 + v)] p_{x_2} - \phi v s p_{x_3} - \phi s p_{u_1} + \phi v s p_{u_2} \\
\vdots \\
-\phi v p_{x_{i-1}} + (r + \phi) p_{x_i} - \phi s p_{u_{i-1}} \\
\phi v s p_{x_0} - \phi s p_{x_1} + \phi s^2 p_{u_0} \\
\phi v s p_{n_1} - \phi s p_{x_2} + (q + \phi s^2) p_{u_1} \\
\phi v s p_{x_2} - \phi s p_{x_3} + (q + \phi s^2) p_{u_2} \\
\vdots \\
q p_{u_i}
\end{pmatrix}
\]

(4.13)

Hence we can determine \( H_{p_x} \) or \( H_{p_u} \) respectively by keeping \( p_{x_i} \) or \( p_{u_i} = 0 \) for each \( i \).

Next we need to find \( \langle p_{x_0}, H_{p_{x_0}} \rangle \) and \( \langle p_{u_0}, H_{p_{u_0}} \rangle \) as in \( \alpha_{x_0} \) and \( \alpha_{u_0} \) respectively, as well as \( \alpha_{x_1}, \alpha_{p_x} \) and \( \alpha_{u_0}, \alpha_{p_u} \) as contained in \( x_i, \alpha_i \) and \( u_i, \alpha_i \) respectively, as well as \( \alpha_{x_1}, \alpha_{p_x} \) and \( \alpha_{u_0}, \alpha_{p_u} \).

From step 3 of the algorithm for DOCP with \( i = 0 \), we have

\[ \alpha_{x_0} = \frac{g_{x_0} \cdot g_{x_0}}{\langle p_{x_0}, H_{p_{x_0}} \rangle} \quad \text{and} \quad \alpha_{u_0} = \frac{g_{u_0} \cdot g_{u_0}}{\langle p_{u_0}, H_{p_{u_0}} \rangle}. \]

(4.14)

From the algorithm again, we deduce,

\[ \langle p_{x_0}, H_{p_{x_0}} \rangle = \phi v^2 p_{x_0} = \phi v^2 \left\| p_{x_0} \right\|^2 = \phi v^2 \left\| g_{x_0} \right\|^2. \]

\[ \langle p_{u_0}, H_{p_{u_0}} \rangle = \phi s^2 p_{u_0} = \phi s^2 \left\| p_{u_0} \right\|^2 = \phi s^2 \left\| g_{u_0} \right\|^2. \]

(4.15)

\[ \alpha_{x_0} = \frac{\left\| g_{x_0} \right\|^2}{\phi v^2 \left\| g_{x_0} \right\|^2} = \frac{1}{\phi v^2}, \quad \text{and} \quad \alpha_{u_0} = \frac{\left\| g_{u_0} \right\|^2}{\phi s^2 \left\| g_{u_0} \right\|^2} = \frac{1}{\phi s^2}. \]

(4.16)

\[ x_1 = x_0 + \alpha_{x_1} p_{x_0} = x_0 - \alpha_{x_1} g_{x_0} = x_0 - \frac{g_{x_0}}{\phi v^2} \quad \text{and} \quad u_1 = u_0 + \alpha_{u_0} p_{u_0} = u_0 - \alpha_{u_0} g_{u_0} = u_0 - \frac{g_{u_0}}{\phi s^2}. \]

(4.17)

When \( i = 1 \), from step 3 of the algorithm, we have

\[ \langle p_{x_0}, H_{p_{x_0}} \rangle = [r + \phi (1 + v)] p_{x_0}^2 \quad \text{and} \quad \langle p_{u_0}, H_{p_{u_0}} \rangle = [q + \phi s^2] p_{u_0}^2. \]

(4.18)

Hence,

\[ \alpha_{x_1} p_{x_0} = \frac{\left\| g_{x_0} \cdot g_{x_0} \right\|}{\langle p_{x_0}, H_{p_{x_0}} \rangle} p_{x_0} = \frac{\left\| g_{x_0} \cdot g_{x_0} \right\|}{[r + \phi (1 + v)] p_{x_0}^2} p_{x_0} = \frac{\left\| g_{x_0} \cdot g_{x_0} \right\|}{[r + \phi (1 + v)] p_{x_0}}. \]

(4.19)

\[ \alpha_{u_0} p_{u_0} = \frac{\left\| g_{u_0} \cdot g_{u_0} \right\|}{\langle p_{u_0}, H_{p_{u_0}} \rangle} p_{u_0} = \frac{\left\| g_{u_0} \cdot g_{u_0} \right\|}{[q + \phi s^2] p_{u_0}^2} p_{u_0} = \frac{\left\| g_{u_0} \cdot g_{u_0} \right\|}{[q + \phi s^2] p_{u_0}}. \]

(4.20)
Using equation (3.5), i.e. \( p_k = -\left(g_k \cdot g_k\right) \sum_{i=0}^{k} \frac{g_i}{\left(g_i \cdot g_i\right)} \), we generate \( p_k \) or \( p_{ki} \) as

\[
p_k = -\left(g_k \cdot g_k \right) \sum_{i=0}^{k} \frac{g_i}{\left(g_i \cdot g_i\right)} = -\left(g_k \cdot g_k \right) \left[ \frac{g_k}{\left(g_k \cdot g_k\right)} + \frac{g_k}{\left(g_k \cdot g_k\right)} \right]
\]

\[
= -g_k \frac{\left(g_k \cdot g_k\right)}{\left(g_k \cdot g_k\right)} g_k
\]

\[
= -g_k \frac{\left(g_k \cdot g_k\right)}{\left(g_k \cdot g_k\right)} \frac{g_k}{\left(g_k \cdot g_k\right)}
\]

\[
\therefore \alpha_n p_k = -\frac{\left(g_k \cdot g_k\right)}{\left[g + \phi(1 + \nu)\right] \left[ -g_k \frac{\left(g_k \cdot g_k\right)}{\left(g_k \cdot g_k\right)} - \frac{g_k}{\left(g_k \cdot g_k\right)} \right]}
\]

\[
= -\frac{\left(g_k \cdot g_k\right)}{\left[g + \phi(1 + \nu)\right] \left[ -g_k \frac{\left(g_k \cdot g_k\right)}{\left(g_k \cdot g_k\right)} - \frac{g_k}{\left(g_k \cdot g_k\right)} \right]}
\]

Similarly

\[
\therefore \alpha_n p_k = -\frac{\left(g_k \cdot g_k\right)}{\left[q + \phi s^2\right] \left[ -g_k \frac{\left(g_k \cdot g_k\right)}{\left(g_k \cdot g_k\right)} - \frac{g_k}{\left(g_k \cdot g_k\right)} \right]}
\]

and so,

\[
x_k = x_i + \alpha_n p_k = x_i + \frac{\left(g_k \cdot g_k\right)}{\left[g + \phi(1 + \nu)\right] \left[ -g_k \frac{\left(g_k \cdot g_k\right)}{\left(g_k \cdot g_k\right)} - \frac{g_k}{\left(g_k \cdot g_k\right)} \right]}
\]

\[
u_k = \nu_i + \alpha_n p_k = \nu_i + \frac{\left(g_k \cdot g_k\right)}{\left[q + \phi s^2\right] \left[ -g_k \frac{\left(g_k \cdot g_k\right)}{\left(g_k \cdot g_k\right)} - \frac{g_k}{\left(g_k \cdot g_k\right)} \right]}
\]

Hence for \( k \geq 2 \), we can generate values for \( x_k \) and \( u_k \) from the expressions

\[
x_k = x_{k-1} + \frac{\left(g_k \cdot g_k\right)}{\left[g + \phi(1 + \nu)\right] \left[ -g_k \frac{\left(g_k \cdot g_k\right)}{\left(g_k \cdot g_k\right)} + \sum_{i=0}^{k-1} \frac{g_k}{\left(g_i \cdot g_i\right)} \right]}
\]

and

\[
u_k = \nu_{k-1} + \frac{\left(g_k \cdot g_k\right)}{\left[q + \phi s^2\right] \left[ -g_k \frac{\left(g_k \cdot g_k\right)}{\left(g_k \cdot g_k\right)} + \sum_{i=0}^{k-1} \frac{g_k}{\left(g_i \cdot g_i\right)} \right]}
\]

Thus we have illustrated that we can analytically find the various parts of the DOCP algorithm and also generate the descent sequence \( z = (x_1, x_2, \ldots, x_k, u_1, u_2, \ldots, u_k) \), which solve our problem.
V. Conclusions

We have successfully presented two approaches for constructing the conjugate search directions $p_{i}^{x}$ and $p_{i}^{u}$ as required in the ECGM algorithm for discrete optimal hcontrol problems (DOCP). We also have shown that the expression for the conjugate search direction $p_{i}^{x}$ is true for all positive integers $n$.

Finally, we have demonstrated that we can find $p_{i}^{x}$, $p_{i}^{u}$, $\alpha_{i}^{x}$, $\alpha_{i}^{u}$ control vector $u_{i}$ and its corresponding trajectory $x_{i}$ by considering a one – dimensional discrete optimal control problem in an attempt to minimize the performance index. Implementing this algorithm either through analytical means or a computer programming language will be less burdensome. We encourage other researchers to look into generating search directions for ECGM algorithm for the case of continuous optimal control problems. Furthermore, re-examine the parameter $\beta_{i} = \left( \frac{g_{i+1}, g_{i+1}}{g_{i}, g_{i}} \right)$, used to update the search directions, with the intent of improving the performance of the algorithm for the continuous optimal control problems.

References