Imbedding the Multiplier in a Discretized Optimal Control Problem
With Real Coefficients Via the Penalty and Multiplier Methods

O. Olotu
Department of Mathematical Science, Federal University of Technology,
PMB 704, Akure, Ondo State, Nigeria

Abstract: Problem statement: Many earlier schemes, particularly the Function Space Algorithm (FSA) which sidetracks the knowledge of operator, for solving quadratic optimal control problems have been computationally involving and iteratively high. Approach: Though, some of these earlier schemes developed operators consisting of complicated integrals still very difficult to evaluate. Here, objectively, a new scheme, Discretized Continuous Algorithm (DCA), is proposed with developed associated operator consisting of a series of summation replacing the integrals of the earlier schemes, thus enhancing much more feasible results and lower iterations. Results: Methodologically, the new scheme uses the penalty-multiplier method to obtain an unconstrained formulation whose bilinear form expression leads to the construction of operator amenable to the Conjugate Gradient Method (CGM). Conclusion/Recommendations: An hypothetical example is considered and results, tabulated per cycle, are more feasible and less iterative than some of the existing methods.

Key words: Penalty, multiplier, quadratic, bilinear form and associated operator

INTRODUCTION

The objective of this research is to develop a Discretized Continuous Algorithm (DCA), using the combination of the penalty and multiplier methods (Rockafellar, 1974), for solving quadratic constrained continuous control problems with real coefficients in order to reduce the cumbersome calculations inherent in earlier schemes such as Function Space Algorithm (FSA) which sidetrack the construction of operator, Extended Conjugate Gradient Method (ECGM) and Imbedded Extended Conjugate Gradient Method (MECGM). The last three did develop operators consisting of a series of integral evaluations uniquely adapted to their schemes. To numerically achieve the development of DCA, we used the finite difference method and discretization of the differential constraint and time interval (Di Pillo and Grippo, 1982; Dennis and Schnable, 1983) respectively to obtain a discretized constrained formulation of the problem. Using Fletcher and Reeves’s method (Fletcher and Reeves, 1964) on function minimization and modified Ibiejugba and Nnumayi’s method (Ibiejugba and Onumanyi, 1984) on function operator, an operator was constructed amenable to the discretized scheme. Consequently, a conjugate gradient method was applied to the constructed operator which serves as a framework for the generalized scheme given in materials and methods.

MATERIALS AND METHODS

Generalized problem 1:

\[
\begin{align*}
\text{Min} & \quad \int_0^T \left( ax^2(t) + bu^2(t) \right) dt \\
\text{Subject to:} & \\
\dot{x}(t) &= cx(t) + du(t), \quad x(0) = x_0, \quad 0 \leq t \leq T \quad (1)
\end{align*}
\]

where, \( x(t), u(t) \in \mathbb{R}, a, b > 0, \) and \( c \) and \( d \) are any contents not necessarily positive.

The constrained problem (1) can be turned into unconstrained problem via the penalty method and the multiplier method (Fletcher and Reeves, 1964; Di Pillo and Grippo, 1982) as:

\[
\langle Z, AZ \rangle_H = \text{Min}_{\langle \nu, \mu, \lambda \rangle} \int_0^T \left[ ax^2(t) + bu^2(t) \\
+ \mu \| \dot{x}(t) - cx(t) - du(t) \|^2 \\
+ \langle \lambda, x(t) - cx(t) - du(t) \rangle \right] dt \quad (2)
\]

where, \( \mu > 0 \) the penalty constant and \( \lambda \) the multiplier (parameter).
Discretization: By discretizing (2) and subdivide \([0,T]\) into \(n\) equal intervals \([t_k,t_{k+1}]\) at meshpoints:

\[
x_0 < x_1 < x_2 < \ldots < x_{n-1} < x_n
\]

where, \(n-1\) is the number of partition points chosen arbitrarily, thus having \((n+1)\) partition points, with \(x_j = j^\ast \Delta t, j = 0, 1, 2, \ldots, n\) and \(\Delta t = \Delta x\) is the fixed length of each subinterval for \(j = k\) or not. By \(j^\ast \Delta t\), it means \(j\) multiplied by \(\Delta t\) let:

\[
t_0 = 0 \text{ and } t_k = \frac{k\Delta t}{n}, \quad k = 1, 2, 3, \ldots, n, \quad t_n = T
\]

\[
x(k) = x_k(t_k), \quad u(k) = u_k(t_k), \quad k = 0, 1, 2, \ldots, n
\]

By Euler’s scheme or finite difference method:

\[
x(k) = \left( x(k+1) - x(k) \right) / \Delta t \quad k = 0, 1, 2, \ldots, n-1
\]

\[
x_k(t_k) = cx_k(t_k) + du_k(t_k)
\]

\[
\left( x(k+1) - x(k) \right) / \Delta t = cx_k(t_k) + du_k(t_k)
\]

We then have the generalized problem (1) in the form:

\[
\min J(x,u,\lambda) = \sum_{k=0}^{n} \sum_{i=1}^{2} \left[ \Delta t \left( ax^2_k(t_k) + bu^2_k(t_k) \right) + \mu (x_{k+1}(t_{k+1}) - \lambda_k(t_k)) \right]
\]

subject to \( \left( x(k+1) - x(k) \right) / \Delta t = cx_k(t_k) + du_k(t_k) \) (4)

\( x(0) = 0 \)

Application of the penalty and multiplier parameters: Applying the penalty function and the multiplier method to (4) by Olotu and and Olorunsola (2006), we have:

\[
\min J(x,u,\lambda) = \sum_{k=0}^{n} \sum_{i=1}^{2} \left[ \Delta t \left( ax^2_k(t_k) + bu^2_k(t_k) \right) + \mu (x_{k+1}(t_{k+1}) - \lambda_k(t_k)) \right]
\]

subject to \( \left( x(k+1) - x(k) \right) / \Delta t = cx_k(t_k) + du_k(t_k) \) (4)

\( x(0) = 0 \)

Using Euler’s scheme (Olotu and and Olorunsola, 2006; Rockafellar, 1974) and simplifying (8), we have the following:

\[
\langle Z_{x_k}(t_k), AZ_{z_k}(t_k) \rangle = \sum_{k=0}^{n} \sum_{i=1}^{2} \left[ \Delta t \left( ax^2_k(t_k) + bu^2_k(t_k) \right) + \mu (x_{k+1}(t_{k+1}) - \lambda_k(t_k)) \right]
\]

subject to \( \left( x(k+1) - x(k) \right) / \Delta t = cx_k(t_k) + du_k(t_k) \) (4)

\( x(0) = 0 \)
Now, we shall state the theorem establishing the operator $A$ with partial proof and for the purpose of simplicity to minimize lengthiness, henceforth, we shall adopt the following notations for the remaining part of the proof:

$$x_i = x_i(t_i), \quad u_i = u_i(t_i), \quad \lambda_i = \lambda_i(t_i) \quad \text{for } i = 1, 2, 3$$

However, for full detail of the proof (Ibiejugba and Onumanyi, 1984; Olomolino et al., 2006), involving penalty method only.

**Theorem 1**: Let the initial guess of the conjugate algorithm be $Z_0(t_i)$ so that:

$$Z_0^T(t_i) = (x_0, u_0, \lambda_0)$$

Then the control operator $u$ associated with the generalized problem satisfying $Az$ is given by:

$$Z_{x_2}^T(t_i) = (x_{x_2}(t_i), u_{x_2}(t_i), \lambda_{x_2}(t_i))$$

**Proof of theorem 1**: Setting $u_{x_2}(t_i) = \lambda_{x_2} = 0$, in (9) and collecting like-terms, we have:

$$\langle Z_{x_2}(t_i), AZ_{x_2}(t_i) \rangle_{11} = \sum_{k=0}^{n} \left[ x_{x_2}(t_k)[\alpha_k + \mu - 2 \Delta_k] + \lambda_{x_2}(t_k) \right]$$

Define:

$$\Omega_1(t_i) = x_{x_2}(t_i)[\alpha_k + \mu - 2 \Delta_k] + \lambda_{x_2}(t_i)$$

And:

$$f_1(t_i) = x_{x_2}(t_i)[\Delta_k - \mu \Delta_k] + \lambda_{x_2}(t_i)$$

So Eq. 10 becomes. Where:

$$\sum_{k=0}^{n} x_{x_2}(t_k)V_{x_2} + x_{x_2}(t_k)\lambda_{x_2}V_{x_2} = x_{x_2}(t_k)(-\Delta_k c + \lambda_{x_2}(t_k)\Delta_k)$$

Define:

$$\Omega_2(t_i) = u_{x_2}(t_k)[\Delta_k m_k - \mu \Delta_k] + \lambda_{x_2}(t_k)$$

And:

$$f_2(t_i) = u_{x_2}(t_k)[\Delta_k m_k - \mu \Delta_k] + \lambda_{x_2}(t_k)$$

Which are continuous functions on $[0,T]$. Now letting Eq. 15 be:

$$\sum_{k=0}^{n} x_{x_2}(t_k)V_{x_2} + x_{x_2}(t_k)\lambda_{x_2}V_{x_2} = x_{x_2}(t_k)(-\Delta_k c + \lambda_{x_2}(t_k)\Delta_k)$$

We have:

$$V_{x_2} = u_{x_2}(t_k)\beta_k$$

$$\Omega_2(t_k) = u_{x_2}(t_k)[\Delta_k m_k]$$

$$f_2(t_k) = u_{x_2}(t_k)[\Delta_k m_k - \mu \Delta_k] + \lambda_{x_2}(t_k)$$

Where, $r_1 = r_2$ with the exception that $f_1 = f_2$ and $\Omega_1 = \Omega_2$ as in Eq. 14.

Again setting $X_{x_2}(t_k) = u_{x_2}(t_k) = 0$ implying that $x_{x_2}(t_k) = 0$ in (7) and collecting like-terms, we have:
\[ \sum_{k=0}^{n} x_k [-2 \lambda_{k2} - \Delta_c] + \dot{x}_k [\lambda_{k2} \Delta_c] + u_k [-\lambda_{k2} \Delta_c] \, da \]  

(20)

Define:

\[ \Omega_2(t_k) = \lambda_{k2} (-2 - \Delta_c) \text{ and } f_2(t_k) = \lambda_{k2} \Delta_c. \]

Now, setting Eq. 7 to:

\[ \sum_{k=0}^{n} x_k [V_{01}] + \dot{x}_k [V_{02}] + u_k [V_{03}] + [0]V_{33} \]

Then we have:

\[ V_{02}(t_k) = -\lambda_{k2} \Delta_c d \text{ and } V_{33}(t_k) = 0 \]

\[ V_{01}(t_k) = -\sinh(T)f_i(0) \]

\[ + \int_0^{T} f_i(s_k) \cosh(s_k - s_k) ds_k - \int_0^{T} \Omega_2(t_k) \sinh(s_k - s_k) ds_k \]  

(21)

+ \Omega_2(0) \cosh(T) + r_i \sinh(T)

where, \( r_i = r_i \) except that \( f_i = f_i \) and \( \Omega_i = \Omega_i \) as in (13).

Having constructed operator \( A \), with entries given above as \( V_{11}, V_{12}, V_{13}, V_{21}, V_{22}, V_{23}, V_{31}, V_{32}, V_{33} \), we now consider Example 1, below, to test the success of the scheme.

Example:

\[ \min \int_0^T (x^2(t) + u^2(t))dt \]

Such that:

\[ X = 2.095x(t) + 1.904u(t) \]

\( \Delta_k = \text{The step size} \)

\( \mu = \text{The penalty constant} \)

\( \lambda = \text{The multiplier} \, a = 1, b = 1, c = 2.095 \text{ and } d = 1.904 \)

We apply QBASIC programming language to execute the developed algorithm, Discretized Continuous Algorithm (DCA).

RESULTS AND DISCUSSION

In Table 1, the numerical problem has an analytic solution, 1.0647, at the optimum. The initial objective functional is 1.1904 and has corresponding constraint satisfaction 8.90625 at the first cycle. The algorithm, DCA, converges much faster in at most 3 iterations. It is observed that the objective value per cycle gets closer to the analytic solution much faster than any of the other methods in the Table 1 for each penalty function and the number of iterations. The DCA, ECGM and MECGM converge at low iterations while the FSA converges at comparatively high iteration in all cases at the 50th iteration. The constraint satisfaction is best with the ECGM algorithm though its computational results deviate from the analytic result.

CONCLUSION

The results, obtained with DCA, show that it is a feasible method that compares favorably with the other existing methods. The convergence profile results reveal that it is a great improvement over the function space Algorithm. The reason is that the knowledge of operator which FSA sidetrack is made available in the new method.

It is recommended that the reasons for the faster convergence of the penalty-multiplier method over any of the other methods can algebraically be examined through the spectrum of the matrix operator associated with the bilinear form expression. It is also recommended that the choice of the optimal value of the penalty function can be obtained by equating the gradient of the Lagrangian to zero.

REFERENCES

Lagrangian function in non-linear programming
http://portal.acm.org/citation.cfm?id=641191
Fletcher, R. and C.M. Reeves, 1964. Function
minimization by conjugate gradient. Comput. J.
Ibiejugba, M.A. and P. Onumanyi, 1984. A control
operator and some of its applications. J. Math Anal.
Appli., 103: 31-47.
Olotu, O. and S.A. Olorunsola, 2006. On the
convergence profile of a discretized scheme for a
two-dimensional constrained optimal control
http://ajol.info/index.php/jonamp/article/view/4017
Omolehin, J.O., K. Rauf, B. Opawoye and W.B. Yahya.
2006. Jacobian approach to optimal determination
of perturbation parameter for gradient method. J.
http://ajol.info/index.php/jonamp/article/view/4014
4/0
Rockafellar, R.T., 1974. Augmented lagrange multiplier
functions and duality in nonconvex programming:
Society for industrial and applied mathematics. J.