

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:

$$\text{Min} \int_0^T (ax^2(t) + bu^2(t)) dt$$

Subject to:

$$\dot{x}(t) = cx(t) + du(t), \quad x(0) = x_0, \quad 0 \leq t \leq T \quad (1)$$

where, $x(t), u(t) \in \mathbb{R}$, $a, b > 0$, 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:

$$\begin{aligned} \langle Z, AZ \rangle_H = \text{Min}_{(x,u,\mu,\lambda)} \int_0^T \{ & ax^2(t) + bu^2(t) \\ & + \mu \|\dot{x}(t) - cx(t) - du(t)\|^2 \\ & + \langle \lambda, \dot{x}(t) - cx(t) - du(t) \rangle \} dt \end{aligned} \quad (2)$$

where, $\mu > 0$ the penalty constant and λ the multiplier (parameter).

Discretization: By discretizing (2) and subdivide [0,T] into n equal intervals $[t_k, t_{k+1}]$ at meshpoints:

$$x_0 < x_1 < x_2 < x_3 < \dots < x_{n-1} < x_n$$

where, n-1 is the number of partition points chosen arbitrarily, thus having (n+1) partition points, with $x_j = j^* \Delta_j$, $j = 0, 1, 2, \dots, n$ and $\Delta_j = \Delta_k$ is the fixed length of each subinterval for $j = k$ or not By $j^* \Delta_j$, it means j multiplied by Δ_j let:

$$t_0 = 0 \text{ and } t_k = \sum_{j=1}^{k-1} \Delta_j, \quad k = 1, 2, 3, \dots, n, \quad t_n = T$$

$$x(k) = x_k(t_k), \quad u(k) = u_k(t_k), \quad k = 0, 1, 2, \dots, n$$

By Euler's scheme or finite difference method:

$$\dot{x}(k) = (x(k+1) - x(k)) / \Delta_k \quad k = 0, 1, 2, \dots, N-1$$

$$\dot{x}_k(t_k) = cx_k(t_k) + du_k(t_k)$$

$$(x(k+1) - x(k)) / \Delta_k = cx_k(t_k) + du_k(t_k) \quad (3)$$

$$\langle \dot{x}_k(t_k), x_{k+1}(t_k) - x_k(t_k) - cx_k(t_k) - du_k(t_k) \rangle, \quad x_0(0) = 0$$

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

$$\min J_{(x,u)} = \sum_{k=0}^n \Delta_k (ax_k^2(t_k) + bu_k^2(t_k))$$

$$\text{subject to } (x(k+1) - x(k)) / \Delta_k = cx_k(t_k) + du_k(t_k) \quad (4)$$

$$x(0) = 0$$

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:

$$\text{Min} J(x, u, \mu, \lambda) = \sum_{k=0}^n \left\{ \begin{array}{l} \Delta_k (ax_k^2(t_k) + bu_k^2(t_k)) + \mu [x_{k+1}(t_{k+1}) \\ -x_k(t_k) - \Delta_k cx_k(t_k) - d\Delta_k u_k(t_k)]^2 \\ + \langle \lambda_k, x_{k+1}(t_{k+1}) - x_k(t_k) - \Delta_k cx_k(t_k) \\ - \Delta_k du_k(t_k) \rangle \end{array} \right\} \quad (5)$$

$$\begin{aligned} &= \sum_{k=0}^n \{ x_k^2(t_k) \alpha_k + u_k^2(t_k) \beta_k + y_k^2(t_k) \mu + n_k x_k(t_k) y_k(t_k) \\ &+ m_k u_k(t_k) y_k(t_k) + p_k x_k(t_k) u_k(t_k) + \lambda_k(t_k) y_k(t_k) \\ &- \lambda_k(t_k) x_k(t_k) - \lambda_k(t_k) \Delta_k cx_k(t_k) - \lambda_k(t_k) \Delta_k du_k(t_k) \} \end{aligned} \quad (6)$$

Where:

$$y_k(t_k) = x_{k+1}(t_{k+1}),$$

$$\alpha_k = \mu + 2\mu \Delta_k c + \Delta_k^2 c^2 \mu + a \Delta_k$$

$$\beta_k = b \Delta_k + \Delta_k^2 d^2 \mu$$

$$n_k = -2\mu - 2\mu \Delta_k c$$

$$m_k = -2\mu \Delta_k d$$

$$p_k = -2\mu \Delta_k d - 2\Delta_k^2 cd$$

The bilinear form expression associated with (6) is given by:

$$\begin{aligned} \langle Z_{k1}(t_k), AZ_{k2}(t_k) \rangle = & \sum_{k=0}^n \{ \alpha_k x_{k1}(t_k) x_{k2}(t_k) + \beta_k u_{k1}(t_k) u_{k2}(t_k) \\ & + y_{k1}(t_k) y_{k2}(t_k) \mu + n_k y_{k1}(t_k) x_{k2}(t_k) + n_k y_{k2}(t_k) x_{k1}(t_k) \\ & + m_k y_{k1}(t_k) u_{k2}(t_k) + m_k y_{k2}(t_k) u_{k1}(t_k) + p_k x_{k1}(t_k) u_{k2}(t_k) \quad (7) \\ & + p_k x_{k2}(t_k) u_{k1}(t_k) + \lambda_{k1}(t_k) y_{k2}(t_k) + \lambda_{k2}(t_k) y_{k1}(t_k) \\ & - \lambda_{k1}(t_k) x_{k2}(t_k) - \lambda_{k2}(t_k) x_{k1}(t_k) - \lambda_{k1}(t_k) x_{k2}(t_k) \Delta_k c \\ & - \lambda_{k2}(t_k) x_{k1}(t_k) \Delta_k c - \lambda_{k1}(t_k) u_{k2}(t_k) \Delta_k d - \lambda_{k2}(t_k) u_{k1}(t_k) \Delta_k d \end{aligned}$$

And

$$\begin{aligned} AZ_{K2}(t_k) &= \begin{pmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{pmatrix} \begin{pmatrix} x_{k2} \\ u_{k2} \\ \lambda_{k2} \end{pmatrix} \\ &= \begin{pmatrix} A_{11} x_{k2} + A_{12} u_{k2} + A_{13} \lambda_{k2} \\ A_{21} x_{k2} + A_{22} u_{k2} + A_{23} \lambda_{k2} \\ A_{31} x_{k2} + A_{32} u_{k2} + A_{33} \lambda_{k2} \end{pmatrix} \end{aligned} \quad (8)$$

where, $z_k(t_k) = (x_k(t_k), u_k(t_k), \lambda_k(t_k))$.

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

$$\begin{aligned} \langle Z_{k1}(t_k), AZ_{k2}(t_k) \rangle_H = & \sum_{k=0}^n \{ \alpha_k x_{k1}(t_k) x_{k2}(t_k) + \beta_k u_{k1}(t_k) u_{k2}(t_k) \\ & + \mu \Delta_k^2 \dot{x}_{k1}(t_k) \dot{x}_{k2}(t_k) - \mu \Delta_k \dot{x}_{k1}(t_k) x_{k2}(t_k) - \mu \Delta_k x_{k1}(t_k) \dot{x}_{k2}(t_k) \\ & + \mu x_{k1}(t_k) x_{k2}(t_k) + n_k x_{k1}(t_k) \dot{x}_{k2}(t_k) \Delta_k - n_k x_{k1}(t_k) x_{k2}(t_k) \\ & + n_k \Delta_k \dot{x}_{k1}(t_k) x_{k2}(t_k) - n_k x_{k2}(t_k) x_{k1}(t_k) + m_k u_{k1}(t_k) \dot{x}_{k2}(t_k) \Delta_k \\ & - m_k x_{k1}(t_k) u_{k2}(t_k) + m_k \Delta_k u_{k2}(t_k) \dot{x}_{k1}(t_k) - m_k u_{k1}(t_k) x_{k2}(t_k) \\ & + p_k u_{k2}(t_k) x_{k1}(t_k) + p_k x_{k2}(t_k) u_{k1}(t_k) + \lambda_{k1}(t_k) \dot{x}_{k2} \Delta_k \\ & - \lambda_{k1}(t_k) x_{k2}(t_k) + \lambda_{k2}(t_k) \dot{x}_{k1} \Delta_k - \lambda_{k2}(t_k) x_{k1}(t_k) \\ & - \lambda_{k1}(t_k) x_{k2}(t_k) - \lambda_{k2}(t_k) x_{k1}(t_k) - \lambda_{k1}(t_k) x_{k2}(t_k) \Delta_k c \\ & - \lambda_{k2}(t_k) x_{k1}(t_k) \Delta_k c - \lambda_{k1}(t_k) u_{k2}(t_k) \Delta_k d - \lambda_{k2}(t_k) u_{k1}(t_k) \Delta_k d \} \end{aligned} \quad (9)$$

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:

$$x_{ki} = x_{ki}(t_k), u_{ki} = u_{ki}(t_k), \lambda_{ki} = \lambda_{ki}(t_k) \text{ for } i=1,2,3$$

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

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

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

Then the control operator a associated with the generalized problem 1.satisfying Az_{k2} is given by:

$$Z_{k2}^T(t_k) = (x_{k2}(t_k), u_{k2}(t_k), \lambda_{k2}(t_k))$$

Proof of theorem 1: Setting $u_{k2}(t_k) = \lambda_{k2} = 0$, in (9) and collecting like-terms, we have:

$$\begin{aligned} \langle Z_{k1}(t_k), AZ_{k2}(t_k) \rangle_H = & \\ \sum_{k=0}^n \{ & x_{k1} [x_{k2} (\alpha_k + \mu - 2n_k) + \dot{x}_{k2} (-\mu \Delta_k + \Delta_k n_k)] \\ & + \dot{x}_{k1} [x_{k2} (\Delta_k n_k - \mu \Delta_k) + \dot{x}_{k2} \mu \Delta_k^2] + u_{k1} [x_{k2} (p_k - m_k) + \dot{x}_{k2} \Delta_k m_k] \\ & + \lambda_{k1} [x_{k2} (-1 - \Delta_k c) + \dot{x}_{k2} \Delta_k] \} \end{aligned} \quad (10)$$

Define:

$$\Omega_1(t_k) = x_{k2} (\alpha_k + \mu - 2n_k) + \dot{x}_{k2} (-\mu \Delta_k + \Delta_k n_k)$$

And:

$$f_1(t_k) = x_{k2} (\Delta_k n_k - \mu \Delta_k) + \dot{x}_{k2} \mu \Delta_k^2$$

So Eq. 10 becomes. Where:

$$\sum_{k=0}^n x_{k1} \cdot V_{11} + \dot{x}_{k1} \cdot \dot{V}_{11} + u_{k1} \cdot V_{21} + \lambda_{k1} \cdot V_{31}$$

$$V_{21} = x_{k2} (p_k - m_k) + \dot{x}_{k2} \Delta_k m_k \quad (11)$$

$$V_{31}(t_k) = x_{k2}(t_k)(-1 - \Delta_k c) + \dot{x}_{k2}(t_k) \Delta_k \quad (12)$$

$$\begin{aligned} V_{11}(t_k) = A_{11}(t_k) = & -\sinh(T)f_1(0) + \int_0^T f_1(s_k) \cosh(t_k - s_k) ds_k \\ & - \int_0^T \Omega_1(t_k) \sinh(T - s_k) ds_k + \Omega_1(t_k) \cosh(T) \quad (13) \\ & + r_1 \sinh(T) \end{aligned}$$

And:

$$\begin{aligned} \tau_1 = & \frac{1}{\sinh(T)} \{-\sinh(T)f_1(0) \\ & + \int_0^T f_1(s_k) \cosh(t_k - s_k) ds_k - \int_0^T \Omega_1(s_k) \sinh(T - s_k) ds_k \\ & - \Omega_1(0) \cosh(T) + \Omega_1(T)\} \end{aligned} \quad (14)$$

In Eq. 7, setting:

$$x_{k2}(t_k) = \lambda_{k2}(t_k) = 0 \rightarrow \dot{x}_{k2}(t_k) = 0$$

And collecting like-terms we have:

$$\begin{aligned} \langle Z_{k1}, AZ_{k2}(t_k) \rangle_H = & \\ \sum_{k=0}^n \{ & x_{k1}(t_k) [u_{k2}(p_k - m_k) + \dot{x}_{k1} (\Delta_k m_k u_{k2})] \\ & + u_{k1} [u_{k2} \beta_k] + \lambda_{k1} [-u_{k2} \Delta_k d] \} \end{aligned} \quad (15)$$

Now define:

$$\Omega_2(t_k = u_{k2}(t_k)(p_k - m_k) \text{ and } f_2(t_k) = u_{k2}(t_k) \Delta_k m_k$$

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

$$\sum_{k=0}^n x_{k1}(t_k) V_{12} + \dot{x}_{k1}(t_k) \dot{V}_{12} + u_{k1}(t_k) V_{22} + \lambda_{k2}(t_k) V_{32} \quad (16)$$

We have:

$$V_{22} = u_{k2}(t_k) \beta_k \quad (17)$$

$$V_{32} = u_{k2}(t_k) (-\Delta_k d) \quad (18)$$

$$\begin{aligned} V_{12}(t_k) = & -\sinh(T)f_2(0) \\ & + \int_0^T f_2(s_k) \cosh(t_k - s_k) ds_k - \int_0^T \Omega_2(t_k) \sinh(T - s_k) ds_k \quad (19) \\ & + \Omega_2(0) \cosh(T) + r_2 \sinh(T) \end{aligned}$$

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

Again setting $X_{k2}(t_k) = u_{k2}(t_k) = 0$ implying that $\dot{x}_{k2}(t_k) = 0$ in (7) and collecting like-terms, we have:

$$\sum_{k=0}^n x_{k1}[-2\lambda_{k2} - \Delta_k c] + \dot{x}_{k1}[\lambda_{k2}\Delta_k] + u_{k1}[-\lambda_{k2}\Delta_k d]a \quad (20)$$

Define:

$$\Omega_3(t_k) = \lambda_{k2}(-2 - \Delta_k c) \text{ and } f_3(t_k) = \lambda_{k2}\Delta_k$$

Now, setting Eq. 7 to:

$$\sum_{k=0}^n x_{k1}[V_{13}] + \dot{x}_{k1}[\dot{V}_{13}] + u_{k1}[V_{23}] + [0]V_{33}$$

Then we have:

$$V_{23}(t_k) = -\lambda_{k2}\Delta_k d \text{ and } V_{33}(t_k) = 0$$

$$V_{13}(t_k) = -\sinh(T)f_3(0) + \int_0^T f_3(s_k) \cosh(t_k - s_k) ds_k - \int_0^T \Omega_3(t_k) \sinh(T - s_k) ds_k + \Omega_3(0) \cosh(T) + r_3 \sinh(T) \quad (21)$$

where, $r_3 = r_1$ except that $f_1 = f_3$ and $\Omega_1 = \Omega_3$ 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:

$$\text{Min} \int_0^1 (x^2(t) + u^2(t)) dt$$

Such that:

$$\dot{X} = 2.095x(t) + 1.904u(t)$$

Δ_k = The step size

μ = The penalty constant

λ = The multiplier $a = 1, b = 1, c = 2.095$ 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,

Table 1: Shows the numerical solutions of all algorithms compared to DCA with analytic solution 1.0647

Penalty/multiplier Parameters	Algorithm	No. of iterations	Objective funct.	Constrained satisfaction
$\mu = s.5, \lambda = -2.88$	DCA	20	1.1904	8.90625
	FSA	50	1.6517	11.62270
	ECGM	7	1.0956	0.45440
	MECGM	10	0.0715	1.12490
$\mu = 1, \lambda = -0.6$	DCA	3	1.5370	11.12500
	FSA	50	1.6251	11.29900
	ECGM	7	1.4834	0.13810
	MECGM	4	0.7073	0.95020
$\mu = 1.5, \lambda = 9.11$	DCA	3	1.1686	8.32750
	FSA	50	1.6001	10.98840
	ECGM	6	1.5570	0.08650
	MECGM	3	0.8686	1.16160
$\mu = 2, \lambda = -10.57$	DCA	2	1.1746	8.36210
	FSA	50	1.5768	10.69020
	ECGM	7	1.4686	0.03530
	MECGM	3	0.9386	1.04770
$\mu = 2.5, \lambda = -10.37$	DCA	3	1.1774	8.37820
	FSA	50	1.5550	10.40200
	ECGM	6	1.5521	8.12620
	MECGM	2	1.0178	1.63130

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

Dennis, J.E. and R.B. Schnable, 1983. Numerical Methods for Unconstrained Optimization and Nonlinear Equations. Princeton-Hall Inc., Englewood Cliffs, New Jersey.

- Di Pillo, G. and L. Grippo, 1982. A new augmented Lagrangian function in non-linear programming problems. *J. Optimiz. Theor. Appl.*, 36: 495-519. <http://portal.acm.org/citation.cfm?id=641191>
- Fletcher, R. and C.M. Reeves, 1964. Function minimization by conjugate gradient. *Comput. J. Elect. Company Lab.*, 7: 149-154.
- Ibiejugba, M.A. and P. Onumanyi, 1984. A control operator and some of its applications. *J. Math Anal. Appl.*, 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 problem. *J. Nigeria Assoc. Math. Phys.*, 10: 537-542. <http://ajol.info/index.php/jonamp/article/view/40172/0>
- Omolehin, J.O., K. Rauf, B. Opawoye and W.B. Yahya. 2006. Jacobian approach to optimal determination of perturbation parameter for gradient method. *J. Nigeria Assoc. Math. Phys.*, 10: 363-370. <http://ajol.info/index.php/jonamp/article/view/40144/0>
- Rockafellar, R.T., 1974. Augmented lagrange multiplier functions and duality in nonconvex programming: Society for industrial and applied mathematics. *J. Control*, 12: 268-285.