Pure and Applied Mathematics Journal
2015; 4(2): 52-56
Published online March 4, 2015 (http://www.sciencepublishinggroup.com/j/pamj)
doi: 10.11648/j.pamj.20150402.14
ISSN: 2326-9790 (Print); ISSN: 2326-9812 (Online)

On application of a new quasi-Newton algorithm for solving optimal control problems

Felix Makanjuola Aderibigbe, Adejoke O. Dele-Rotimi, Kayode James Adebayo

Department of Mathematical Sciences, Ekiti State University, Ado Ekiti, Nigeria

Email address:
fm_aderibigbe@yahoo.com.uk (F. M. Aderibigbe), delerotimi@ymail.com (A. O. Dele-Rotimi), akj7070@yahoo.com (K. J. Adebayo)

To cite this article:
Felix Makanjuola Aderibigbe, Adejoke O. Dele-Rotimi, Kayode James Adebayo. On Application of a New Quasi-Newton Algorithm for Solving Optimal Control Problems. *Pure and Applied Mathematics Journal*. Vol. 4, No. 2, 2015, pp. 52-56.
doi: 10.11648/j.pamj.20150402.14

Abstract: In this work, we review the construction of the linear operator associated with a class of linear regulator problems subject to the state differential equation. The associated linear operator is then utilized in the derivation of a New Quasi-Newton Method (QNM) for solving this class of optimal control problems. Our results show an improvement over the Classical Quasi-Newton Method.

Keywords: Optimal Control Problem, Classical Quasi-Newton Method, New Quasi-Newton Method, Control Operator

1. Introduction

Optimal control theory is playing an increasingly important role in the design of modern systems. More specifically, control problems play an important role in aerospace, as well as in other applications, where for example, temperature, pressure, and other variables must be kept at desired values regardless of disturbances.[5]. In this paper, we propose a New Quasi-Newton method by adopting the quasi-Newton method algorithm to obtain the solution of the following scalar, linear, optimal control problem of the form:

Problem A:

$$\min_{(x,u)} \int_0^T \{Ax^2(t) + Bu^2(t)\}dt \tag{1.1}$$

Subject to the constraint

$$\dot{x}(t) = Cx(t) + Du(t);\ \ 0 \le t \le T \tag{1.2}$$

$$x(0) = x_0 \tag{1.3}$$

where x is the $n \times 1$ state vector, u is the $q \times 1$ control vector, C is $n \times n$ constant matrix, D is $n \times q$ constant matrix, A and B are symmetric positive definite constant square matrices of dimensions n and q respectively. $\dot{x}(t)$, the derivative of the state, $x(\cdot)$, with respect to time.

As conventional with penalty function techniques, (1.1) to (1.3) may equivalently be written in the form:

$$\min_{(x,u)} \int_0^T \{Ax^2(t) + Bu^2(t) + \mu||\ \dot{x}(t) - Cx(t) + Du(t)||^2\}\,dt \tag{1.4}$$

where $\mu > 0$, is the penalty parameter and $\mu||\ \dot{x}(t) - Cx(t) + Du(t)||^2$ is the penalty term. We assume that μ is a suitably chosen parameter to ensure good constraint satisfaction for each of the problems under consideration. The validity of this claim can be seen from a number of theoretical and numerical results in $[2, pp\ 61-85]$. In here, we seek to apply the quasi-Newton method algorithm to (1.4).

Based on (1.4), application of the QNM algorithm requires that an operator, $\hat{Q}$, be determined such that;

$$\langle y, \hat{Q}y\rangle_H \cong \int_0^T \{Ax^2(t) + Bu^2(t) + \mu\|\dot{x}(t) - Cx(t) + Du(t)\|^2\}\,dt \tag{1.5}$$

where $y^T(t) = (x(t), u(t))$ and $\mathcal{H}$ is a suitably chosen Hilbert space.

According to [8], the operator $\hat{Q}$ is such that

$$(\hat{Q}z)(t) \equiv \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{bmatrix}\begin{bmatrix} x(t) \\ u(t) \end{bmatrix} = \begin{bmatrix} (Q_{11}x)(t) + (Q_{12}u)(t) \\ (Q_{21}x)(t) + (Q_{22}u)(t) \end{bmatrix} \tag{1.6}$$

and the composite units of the linear operator $(Q_{11}x)(t)$, $(Q_{12}u)(t), (Q_{21}x)(t)\ and\ (Q_{22}u)(t)$ are given by:

$$(Q_{11}x)(t) = -\mu[\dot{x}(0) - Cx(0)]Sinh(\mathrm{t})$$

$$+\mu\int_0^t[\dot{x}(s) - Cx(s)]Cosh(t-s)ds - \int_0^t[(A+\mu C^2)x(s) - \mu C\dot{x}(s)]\,Sinh(t-s)[(a+\mu C^2)x(0) - \mu C\dot{x}(0)]Cosh(t) +$$

$$\frac{Sinh(t)}{Sinh(T)}\{(a+\mu C^2)x(T)-\mu C\dot{x}(T)+\mu Sinh(T)[\dot{x}(0)-Cx(0)]-\mu\int_0^T[\dot{x}(s)-Cx(s)]Cosh(T-s)ds+\int_0^T[(A+\mu C^2)x(s)-\mu C\dot{x}(s)]Sinh(T-s)ds$$

$$-[(A+\mu C^2)x(0)-\mu C\dot{x}(0)]Cosh(T)\},\ 0\le t\le T \quad (1.7)$$

$$(Q_{21}x)(t)=\mu CDx(t)-\mu D\dot{x}(t),\ 0\le t\le T \quad (1.8)$$

$$(Q_{12}u)(t)=\mu Du(0)Sinh(t)-\mu\int_0^t Du(s)Cosh(t-s)ds+\mu\int_0^t CDu(s)Sinh(t-s)ds+\mu CDu(0)Cosh(t)+\frac{Sinh(t)}{Sinh(T)}\{\mu CDu(T)-\mu Du(0)Sinh(T)+\mu\int_0^T Du(s)Cosh(T-s)ds+\mu\int_0^T CDu(s)Sinh(T-s)ds+\mu CDu(0)Cosh(T)\}\quad 0\le t\le T \quad (1.9)$$

$$(Q_{22}u)(t)=Bu(t)+\mu D^2u(t),\ 0\le t\le T \quad (1.10)$$

where $(Q_{11}x)(t),(Q_{12}x)(t)$, $(Q_{21}x)(t)$ and $(Q_{22}x)(t)$ are respectively given by (1.7) – (1.10). Reader should see [7] for the proof.

2. Classical Quasi-Newton Algorithm

Nonlinear problems in finite dimensions are generally solved by iteration.[8], for the minimization problem, and [7], for systems of equations, introduced new methods which although iterative in nature, were quite unlike any others in use at the time. These papers together with very important modification and classification of Davidon's work by [4] have sparked a large amount of research in the late sixties and early seventies.

This work has led to a new class of algorithm which has been called by the names quasi-Newton or modification methods. The methods have proved themselves in dealing with systems of *n* equations in *n* unknowns, and the unconstrained minimization of functionals. [3]

The basic idea behind any quasi-Newton method is to eliminate computation of the Hessian in every iteration and the methods are based on Newton's method to find the stationary point of a function, where the gradient is zero or near zero. The Hessian is updated by analyzing successive gradient vectors instead. Detailed overviews of quasi-Newton methods are presented in [8] and [7]. The search direction is obtained by solving

$$B_kp=-\nabla f(x_k)$$

That is, from the Newton's equation but with the Hessian replaced by B_k, a positive definite matrix. Quasi-Newton methods require only the gradient of the objective function to be supplied at each iterate.[8]. Since second derivatives are not required, quasi-Newton methods are more efficient than Newton's method and display a super linear rate of convergence.

In place of the true Hessian $\nabla^2f(x_k)$, they use an approximation B_k, which is updated each step to take account of the additional knowledge gained during the step. The various quasi- Newton methods differ in the choice of B_k. The first quasi-Newton method is the DFP which was soon superceeded by BFGS.

The BFGS method named after Broyden, Fletcher, Goldfarb and Shanno who discovered it in 1970. It is numerically stable and has a very effective "self-correcting properties" account for its superior performance in practice [9]. If the matrix H_k incorrectly estimates the curvature in the objective function, and if this bad estimate slows down the iteration, then the Hessian approximation will tend to correct itself within a few steps.

Since the search direction $p=-H_k\nabla f(x_k)$, this has the advantage that we don't need to solve a linear system to get the search direction, but only do a matrix/vector multiply.

The BFGS update formula is as follows:

$$H_{k+1}=H_k+\frac{y_ky_k^T}{y_k^Ts_k}-\frac{H_ks_ks_k^TH_k}{s_k^TH_ks_k}$$

By taking the inverse, the BFGS update formula for $B_{k+1}(i.e\ H_{k+1}^{-1})$ is obtained:

$$H_{k+1}^{-1}=B_{k+1}=B_k+\left(\frac{1+y_k^TB_ky_k}{y_k^Ts_k}\right)\frac{s_ks_k^T}{s_k^Ty_k}-\frac{s_ky_k^TB_k+B_ky_ks_k^T}{y_k^Ts_k}$$

The BFGS, preserve positive definiteness of the Hessian approximations if and only if $y_k^Ts_k>0$.(see theorem 1 for the proof)

Theorem 1. Let B_i be a symmetric positive-definite matrix, and assume that B_{i+1} is obtained from B_i using the BFGS update formula. Then B_{i+1} is positive definite if and only if $y_i^Ts_i>0$.[9]

Proof. If B_i is positive definite, then it can be factored as $B_i=LL^T$ where L is a nonsingular matrix (Cholesky factorization of B_i). If this factorization is substituted into the BFGS formula for B_{i+1}, then

$$B_{i+1}=LWL^T$$

Where $=I-\frac{\hat{s}\hat{s}^T}{\hat{s}^T\hat{s}}+\frac{\hat{y}\hat{y}^T}{\hat{y}^T\hat{y}}$, $\hat{s}=L^Ts_i$, and $\hat{y}=L^{-1}y_i$ B_{i+1} will be positive definite if and only if W is. To test if W is positive definite, we test if $v^TWv>0$ for all $v\neq 0$. let θ_1 be the angle between v and $\hat{s}$, θ_2 the angle between v and $\hat{y}$, and θ_3 is the angle between $\hat{s}$ and $\hat{y}$. Then

$$v^TWv=v^Tv-\frac{(v^T\hat{s})^2}{\hat{s}^T\hat{s}}+\frac{(v^T\hat{y})^2}{\hat{y}^T\hat{s}}$$

$$=\|v\|^2-\frac{\|v\|^2\|\hat{s}\|^2cos^2\theta_1}{\|\hat{s}\|^2}-\frac{\|v\|^2\|\hat{y}\|^2cos^2\theta_2}{\|\hat{y}\|.\ \|\hat{s}\|cos\theta_3}$$

$$=\|v\|^2\left[1-cos^2\theta_1+\frac{\|\hat{y}\|cos^2\theta_2}{\|\hat{s}\|cos\theta_3}\right]$$

$$=\|v\|^2\left[sin^2\theta_1+\frac{\|\hat{y}\|cos^2\theta_2}{\|\hat{s}\|cos\theta_3}\right]$$

If $y_i^Ts_i>0$, then $\hat{y}^T\hat{s}>0$ and $cos\theta_3>0$; hence $v^TWv>0$ and W is positive definite. If $y_i^Ts_i<0$, then $cos\theta_3<0$; in this case, v can be chosen so that $v^TWv<0$ and so W is not positive definite. This completes the proof.

Given a quadratic functional

$$F(x) = F_0 + \langle a, x\rangle_H + \langle x, Ax\rangle_H$$

For x, a in Hilbert space with A being a positive, symmetric linear operator. The quasi-Newton algorithm is described in the following steps:

Step 1: Guess the initial element, x_0

Step 2: Compute the gradient, g_0

Step 3: Compute the descent direction, $p_0 = -B_0 g_0$; $B_0 = I$

Step 4: Compute the step length, $\alpha_i = \frac{g_k^T g_k}{p_k^T A p_k}$

Step 5: Update the descent sequence, $x_{k+1} = x_k + \alpha_k p_k$

Step 6: Update the gradient $g_{k+1} = \nabla \mathrm{f}(\mathrm{x_{k+1}})$

Step 7: Test for convergence $f(x_k)$ *and* $\|g_k\|$

Step 8: Determine the vector updates

$$s_k = x_{k+1} - x_k$$

$$y_k = \nabla f(x_{k+1}) - \nabla f(x_k)$$

Step 9: compute the new Hessian approximate

$$B_{k+1} = B_k + \left(\frac{1+y_k^T B_k y_k}{y_k^T s_k}\right)\frac{s_k s_k^T}{s_k^T y_k} - \frac{s_k y_k^T B_k + B_k y_k s_k^T}{y_k^T s_k}$$

Step 10: compute the next descent direction

$$p_{k+1} = -B_{k+1}\nabla f(x_{k+1})$$

Step 11: return to step 3

In the iterative steps 2 to 10 above, p_k denotes the descent direction at the k^{th} step of the algorithm, α_k denotes the step length of the descent sequence $\{x_k\}$, I_0 denotes the identity $(n \times n)$ matrix, f denotes the objective function, A is the linear operator, s_k is the difference between two consecutive variable values, y_k is the difference between two consecutive gradient values and g_k denotes the gradient of f at x_k.

3. A New Quasi-Newton Algorithm

The proposed New Quasi-Newton algorithm for solving the control problem (1.1) is as follows:

Step 1: *Guess the initial elements*, x_0, u_0

Step 2: *Compute the gradients*: $g_{0,x}; g_{0,u}$

Step 3: *Compute the descent directions*

$$P_{0,x} = -B_{0,x} g_{0,x}\ P_{0,u} = -B_{0,u} g_{0,u}\ ;\ B_0 = I$$

Step 4: *Compute the step lenghts*:

$$\alpha_{i,x} = \frac{g^T{}_{i,x} g_{i,x}}{p^T{}_{i,x} Q p_{i,x}},\ \alpha_{i,u} = \frac{g^T{}_{i,u} g_{i,u}}{p^T{}_{i,u} Q p_{i,u}}$$

Step 5: *Update the descent sequences*

$$x_{i+1} = x_i + \alpha_{i,x} p_{i,x}\ , u_{i+1} = u_i + \alpha_{i,u} p_{i,u}$$

Step 6: *Update the gradients*

$$g_{i+1} = \nabla f[x_{i+1}(t), u_{i+1}(t), t]$$

Step 7: *Test for convergence with*

$$f(x_i(t), u_i(t), t)\ and\ \|g_i\|$$

Step 8: *Determine the vector updates*

$$s_{i,x} = x_{i+1} - x_i\ , \qquad s_{i,u} = u_{i+1} - u_i$$

$$y_{i,x} = \nabla f(x_{i+1}) - \nabla f(x_i)\ ,\ \ y_{i,u} = \nabla f(u_{i+1}) - \nabla f(u_i)$$

Step 9: *Compute the new Hessian approximates*

$$B_{i+1,x} = B_{i,x} + \left[\frac{1+y^T{}_{i,x} B_i y_{i,x}}{y^T{}_{i,x} s_{i,x}}\right]\left[\frac{s_{i,x} s^T{}_{i,x}}{s^T{}_{i,x} y_{i,x}}\right] - \left[\frac{s_{i,x} y^T{}_{i,x} B_i + B_i y_{i,x} s^T{}_{i,x}}{y^T{}_{i,x} s_{i,x}}\right]$$

$$B_{i+1,u} = B_{i,u} + \left[\frac{1+y^T{}_{i,u} B_i y_{i,u}}{y^T{}_{i,u} s_{i,u}}\right]\left[\frac{s_{i,u} s^T{}_{i,u}}{s^T{}_{i,u} y_{i,u}}\right] - \left[\frac{s_{i,u} y^T{}_{i,u} B_i + B_i y_{i,u} s^T{}_{i,u}}{y^T{}_{i,u} s_{i,u}}\right]$$

Step 10: *Compute the next descent directions*

$$p_{i+1,x} = -B_{i+1,x}\nabla f(x_{i+1,x})\ ,\ \ p_{i+1,u} = -B_{i+1,u}\nabla f(x_{i+1,u})$$

Step 11: *Return to step* 4

In the iterative steps 2 to 10 above,p_i denotes the descent direction at the i^{th} step of the algorithm, α_i denotes the step length of the descent sequence $\{x_i\}$, I denotes the identity $(n \times n)$ matrix, f denotes the objective function, Q is the control operator, s_i is the difference between two consecutive variable values, y_i is the difference between two consecutive gradient values and g_i denotes the gradient of f at x_i. Based on the seventh step of the QNM in solving this class of problems, the following can be used to set the stopping conditions:

1. The function is said to have converged when the gradient value is zero. In order words, all other things as from that point become tend to zero.
2. The Gradient Norm can also be used as the stopping criterion to determine the convergence of the function as the gradient norm tends towards zero.
3. The result can be compared with the analytical results or existing results using other methods.

Note: Two or more of these will be used to determine the convergence of the problems to be tested herein.

4. Test Problems

We now compare performance of our New Quasi-Newton method (NQNM) with that of the Classical Quasi-Newton Method (CQNM) on a number of control problems. The following problems were used (throughout, μ is the penalty constant which may be randomly chosen):

P1. $\min_{(x,u)} \int_0^1 \{x^2(t) + u^2(t)\}\, dt;\ 0 \le t \le 1$

Subject to $\dot{x}(t) = 3x(t) + 2u(t);\ 0 \le t \le 1$

$x(0) = 1$

In an unconstrained form, this becomes

$$min_{(x,u)} \int_0^1 \{x^2(t) + u^2(t) + \mu\|\dot{x}(t) - 3x(t) + 2u(t)\|^2\} dt$$

P2. $min_{(x,u)} \int_0^1 \{2x^2(t) + 2u^2(t)\} dt$

Subject to $\dot{x}(t) = 3x(t) + u(t); 0 \leq t \leq 1$

$x(0) = 1$

which in unconstrained form reduces to

$$min_{(x,u)} \int_0^1 \{2x^2(t) + 2u^2(t) + \mu\|\dot{x}(t) - 3x(t) + u(t)\|^2\}dt$$

P3. $min_{(x,u)} \int_0^1 \{0.5x^2(t) - u^2(t)\}dt$

subject to $\dot{x}(t) = 3x(t) + 2u(t); \quad 0 \leq t \leq 1$

$x(0) = 1$

Or $min_{(x,u)} \int_0^1 \{0.5x^2(t) - u^2(t) + \mu\|\dot{x}(t) + 3x(t) - 2u(t)\|^2\}dt$

P4. $min_{(x,u)} \int_0^1 \{-x^2(t) - u^2(t)\}dt$

subject to $\dot{x}(t) = 3x(t) + 2u(t); \quad 0 \leq t \leq 1$

$x(0) = 1$

Or $min_{(x,u)} \int_0^1 \{x^2(t) + u^2(t) + \mu\|\dot{x}(t) - 3x(t) + 2u(t)\|^2\}dt$

The computational results of the problems in P1-P4 are discussed hereunder using the gradient norm values of the CQNM and the NQNM as the bases of comparison. We present our computational results for the above problems in Tables 1-4 below while a general discussion of the results follow thereafter.

Table 1. *Computational Results for Problem (P1)*

Classical Quasi-Newton Algorithm (CQNM)				**New Quasi-Newton Algorithm (NQNM)**			
Itrn.	X	U	Gradient Norm	Itrn.	X	U	Gradient Norm
0	1	1	179.799889	0	1	1	179.799889
1	0.40577413	1.28610875	33.0453725	1	-1.07214044	1.99769725	319.343042
2	0.40460426	1.28693977	32.9403232	2	-1.74013304	0.61032799	9.60106989
10	0.38930823	1.30003421	31.8162596	10	-1.79970474	0.64500731	0.140696e-3
490	0.29664104	1.32436588	0.51684382e-6	14	-1.79970557	0.64500779	0.51583707e-6

Table 2. *Computational Results for Problem (P2)*

Classical Quasi-Newton Algorithm (CQNM)				**New Quasi-Newton Algorithm (NQNM)**			
Itrn.	X	U	Gradient Norm	Itrn.	X	U	Gradient Norm
0	1	1	250.343764	0	1	1	250.343764
1	0.74056277e-1	1.21251167	13.6915398	1	-0.514307	1.3475559	98.3981051
2	0.73787764e-1	1.21258203	13.5804814	2	-0.7929404	0.13370078	8.3689144
10	0.71516393e-1	1.21321725	13.1857362	10	-0.95077218	0.27507636	0.20949144e-1
14659	-0.16148132e-1	0.79296301	0.19145828e-6	24	-0.95105385	0.27529469	0.19177078e-6

Table 3. *Computational Results for Problem (P3)*

Classical Quasi-Newton Algorithm (CQNM)				**New Quasi-Newton Algorithm (NQNM)**			
Itrn.	X	U	Gradient Norm	Itrn.	X	U	Gradient Norm
0	1	1	2.23606798	0	1	1	2.23606798
1	0.99234303	1.01531394	0.26605812	1	1.01036537	0.97926925	0.92971534
2	0.99239681	1.01514268	0.2678443	2	-0.7929404	0.13370078	8.3689144
10	0.99234086	1.01488661	0.28612792	10	-0.95077218	0.27507636	0.20949144e-1
				24	0.99797326	0.97871204	0.31102406e-6

Table 4. *Computational Results for Problem (P4)*

Classical Quasi-Newton Algorithm (CQNM)				**New Quasi-Newton Algorithm (NQNM)**			
Itrn.	X	U	Gradient Norm	Itrn.	X	U	Gradient Norm
0	1	1	178.011236	0	1	1	178.011236
1	0.41753686	1.30229099	29.4141379	1	-1.34458504	2.21680996	366.851474
2	0.41669374	1.30289397	29.3126686	2	-2.05812417	0.84194188	13.338083
10	0.40765154	1.31064779	28.2702834	10	-1.99074176	0.80447785	0.3983206e-5
				12	-1.99074174	0.80447784	0.95184684e-7

5. Conclusion

From the table of results for P1, it is observed that at the 14th iteration, the terminating norm of the gradient is 0.51583707e-6 using the NQNM while the closest value to it using the CQNM is at the 490th iteration which signifies a significant improvement over the CQNM. From the table of results for P2, it is observed that at the 24th iteration, the gradient norm is 0.19177078e-6 using the NQNM while the closest value to it using the CQNM is at the 14659th iteration which also signifies an improvement over the CQNM. From the tables of results for P3 and P4, it is seen that the closest values to that of the 24th and 12th respectively using the NQNM could not be reached using the CQNM. Hence, the introduction of the control operator which was constructed by [7] into the classical quasi-Newton algorithm has improved the convergence profile. Since the efficiency of the CGM rest on the operator, this justifies the use of the operator in the CQNM as to improve the performance of the CQNM.

Reference

[1] Broyden, C. G., (1965). "A Class of Methods for Solving Nonlinear Simultaneous Equations", Math. Comp., 19, pp. 577-593.

[2] Davidon, W. C,. (1959). "Variable Metric Method for Minimization", Rep. ANL-5990 Rev, Argonne National Laboratories, Argonne, I11

[3] Dennis, J. E. JR and More, J. J., (1977). "Quasi Newton Methods, Motivation and Theory", SIAM Review 19(1), pp 46-89

[4] Fletcher, R., and Powell, M. J. D., (1963), "A rapidly Convergent Descent Method for Minimization." Comp. J., 6, pp 163-168.

[5] George, M. S. (1996), An Engineering Approach To Optimal Control And Estimation Theory. John Wiley & Sons, Inc.

[6] Ibiejugba, M. A. and Onumanyi, P., (1984), "A Control Operator and Some of its Applications," Journal of Mathematical Analysis and Applications, Vol. 103, No.1, pp 31-47.

[7] Ibiejugba, M. A., (1980), "Computing Methods in Optimal Control," Ph. D. Thesis, University of Leeds, Leeds, England.

[8] Igor, G., Stephen, G. N. and Ariella, S., (2009), Linear and Nonlinear Optimization, 2nd edition, George Mason University, Fairfax, Virginia, SIAM, Philadelphia.

[9] Nocedal, J. and Wright, S. J., (2006), Numerical Optimization. 2nd edition. Springer-Verlag, New York.