

Yen, V. and Nagurka, M.L., "A Control Parameterization Approach for Linear Quadratic Systems," Proceedings of the 1989 American Control Conference, Pittsburgh, PA, June 21-23, 1989, pp. 774-779.

A CONTROL PARAMETERIZATION APPROACH FOR LINEAR QUADRATIC SYSTEMS

Vincent Yen Mark L. Nagurka

Department of Mechanical Engineering and The Robotics Institute
Carnegie Mellon University, Pittsburgh, PA 15213

ABSTRACT

This paper develops a control parameterization approach for determining the (near) optimal trajectories of linear time-invariant systems with quadratic performance indices. In solving the linear quadratic (LQ) problem for time-invariant systems each control variable is represented by a set of approximating functions with unknown coefficients. This converts the LQ problem into an unconstrained quadratic programming problem which can be solved for the (near) optimal control parameter values (i.e., the unknown coefficients) by solving a system of linear algebraic equations. As verified by simulation studies, the control parameterization approach is particularly efficient when applied to minimum energy problems and to problems with significantly fewer control variables than state variables.

INTRODUCTION

In addition to variational methods (such as the Hamilton-Jacobi approach) and dynamical programming approaches, mathematical programming techniques represent an alternative methodology for solving optimal control problems. These techniques usually approximate the state and/or control variables by a set of prespecified functions, whose coefficients characterize the state and/or control trajectories. The original problem is thus converted into an algebraic optimization problem, which is solved for the (near) optimal values of the state and/or control parameters that minimize the optimization objective function. A survey of work reported in this area prior to 1970 can be found in [1]; a survey to 1980 can be found in [2]. A more recent study of control parameterization can be found in [3] where optimal control problems with general constraints are addressed.

A direct application of mathematical programming is to characterize the state and control trajectories by their values at a number of prespecified locations (e.g., a number of equally-spaced points in time). The theoretical and practical aspects of this method are covered in [4] for discrete-time systems and in [5] for continuous-time systems.

Ghonaimy and El-Zorkani [6] developed a control parameterization approach for linear quadratic (LQ) problems with free and fixed terminal states. In their approach, each control variable is approximated by a polynomial or finite number of orthogonal functions whose coefficients are sought. Results from a numerical example indicate that satisfactory results can be obtained using polynomials of low order (i.e., less than or equal to four). Although this approach is mathematically elegant, Ghonaimy and El-Zorkani do not discuss the computational requirements of their approach nor compare execution times relative to Riccati equation solvers.

Yen and Nagurka [7] have presented a more recent application of a mathematical programming technique to solve LQ problems. In [7] each state variable is represented by the sum of a polynomial and a finite term Fourier-series. Consequently, this state parameterization approach converts an optimal control problem into an unconstrained quadratic programming problem. Simulation results demonstrate that the method is much more efficient than a Riccati equation solver in handling high order systems.

Based on a similar idea of mathematical programming, this paper develops a control parameterization approach for LQ problems with free and fixed terminal states. Similar to the state parameterization approach developed in [7], this approach converts an unconstrained LQ problem into an unconstrained quadratic programming problem. In contrast to the earlier work, this approach is particularly efficient in handling systems whose number of control variables is significantly less than the order of the system. In addition, this control parameterization approach is a very efficient tool in handling minimum energy problems.

METHODOLOGY

Unconstrained LQ Problems

This paper considers the optimal control of linear time-invariant systems with quadratic performance indices. Such systems can be represented by the state space model:

$$\dot{x} = Ax(t) + Bu(t) \quad (1)$$

with initial condition

$$x(0) = x_0 \quad (2)$$

Here, x is an $N \times 1$ state vector, u is an $L \times 1$ control vector, A is an $N \times N$ system matrix, and B is an $N \times L$ control matrix. The performance index is a quadratic functional defined as:

$$PI = PI_A + PI_B + PI_C \quad (3)$$

where

$$PI_A = \frac{1}{2} x^T(t_f) H x(t_f) \quad (4)$$

$$PI_B = \frac{1}{2} \int_0^{t_f} x^T Q x dt \quad , \quad PI_C = \frac{1}{2} \int_0^{t_f} u^T R u dt \quad (5,6)$$

It is assumed that H and Q are real, symmetric, positive-semidefinite matrices and R is a real, symmetric, positive-definite matrix. The total time interval is $[0, t_f]$, where t_f is the final time. Superscript T denotes transpose. The design objective is to compute the optimal control $u(t)$ and the corresponding optimal trajectory $x(t)$ such that the performance index PI is minimized.

The first step of this control parameterization approach is to divide $[0, t_f]$ into J equal time intervals $[t_0, t_1], [t_1, t_2], \dots, [t_{j-1}, t_j]$, where $t_0=0$ and $t_j = t_f$. The time of each interval is denoted by Δt . It is assumed that each of the L control variables has a single value in each interval $[t_{j-1}, t_j], j = 1, \dots, J$, and is piecewise-continuous in $[0, t_f]$.

Since over each time interval the control variables are single-valued, the system can be represented by the following discrete-time model:

$$x(j+1) = \phi x(j) + \Gamma u(j) \quad \text{for } j = 0, \dots, J-1 \quad (7)$$

where the state transition matrix ϕ and the control matrix of the discrete model Γ are

$$\phi = e^{A\Delta t}, \quad \Gamma = \int_0^{\Delta t} e^{A t} dt B \quad (8,9)$$

and where, for the sake of simplicity, $x(t_j)$ and $u(t_j)$ have been denoted by $x(j)$ and $u(j)$, respectively. From multiple applications of equation (7), the following equation can be obtained:

$$x(j) = \phi x(0) + \sum_{i=0}^{j-1} \phi^{j-i-1} \Gamma u(i) \quad \text{for } j = 1, \dots, J \quad (10)$$

Based on equation (10), the relationship between the state trajectory and control trajectory can be established as:

$$x^* = \phi^* x(0) + \Gamma^* u^* \quad (11)$$

where

$$x^* = \begin{bmatrix} x(1) \\ x(2) \\ \vdots \\ x(J) \end{bmatrix}_{(JN \times 1)}, \quad u^* = \begin{bmatrix} u(0) \\ u(1) \\ \vdots \\ u(J-1) \end{bmatrix}_{(JL \times 1)} \quad (12,13)$$

$$\phi^* = \begin{bmatrix} \phi \\ \phi^2 \\ \vdots \\ \phi^J \end{bmatrix}_{(JN \times N)}, \quad \Gamma^* = \begin{bmatrix} \Gamma & 0 & \dots & 0 \\ \phi \Gamma & \Gamma & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \phi^{j-1} \Gamma & \phi^{j-2} \Gamma & \dots & \Gamma \end{bmatrix}_{(JN \times JL)} \quad (14,15)$$

Here, u^* represents the time history of the quantized control vector and x^* is the corresponding response of the state vector. The subscripts in the parentheses represent the dimensions of the matrices. In equation (11), the term $\phi^* x(0)$ represents the contribution of the initial condition and the term $\Gamma^* u^*$ represents the influence of the control history vector u^* on the system response. Equation (11) will be used to convert the performance index into a function of u^* .

From equations (4), (5), and (6), the three components of the performance index can be written in the following way:

$$PI_A = \frac{1}{2} x^T(J) H x(J), \quad PI_B = PI'_B = \frac{1}{2} \sum_{j=0}^J x^T(j) Q x(j) \Delta t \quad (16,17)$$

$$PI_C = \frac{1}{2} \sum_{j=0}^{J-1} u^T(j) R u(j) \Delta t \quad (18)$$

In equation (17), PI_B , i.e., the state integral part of the performance index, can be approximated using Simpson's rule giving

$$Q_j = \begin{cases} \frac{1}{3} Q & \text{for } j = 0 \text{ and } j = J \\ \frac{2}{3} Q & \text{for } j = 2, 4, \dots, J-2 \\ \frac{4}{3} Q & \text{for } j = 1, 3, \dots, J-1 \end{cases} \quad (19)$$

where J is assumed to be an even number. Note that PI'_B is only an approximation of PI_B . However, PI_C can be computed without approximation from equation (18) since the control vector is assumed to be single-valued at every time interval. For the same reason, $x(J)$ can be computed directly from equation (10) enabling PI_A to be obtained from equation (16) without approximation. Thus the performance index PI can be represented by PI' , defined as

$$PI' = PI_A + PI'_B + PI_C = \frac{\Delta t}{2} [u^{*T} R^* u^* + x^{*T} Q^* x^* + x^T(0) Q_0 x(0)] \quad (20)$$

where

$$R^* = \begin{bmatrix} R & 0 \\ 0 & R \end{bmatrix}_{(LJ \times LJ)}, \quad Q^* = \begin{bmatrix} Q_1 & 0 & \dots & 0 \\ 0 & Q_2 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & 0 & Q_{J-1} & \dots \\ 0 & \dots & 0 & 0 & Q_J + H \end{bmatrix}_{(JN \times JN)} \quad (21,22)$$

By substituting equation (11) into equation (20), the converted performance index can be rewritten as:

$$PI' = \frac{\Delta t}{2} [u^{*T} \Lambda u^* + 2 u^{*T} \Omega x_0 + x_0^T \Psi x_0] \quad (23)$$

where

$$\Lambda = R^* + \Gamma^{*T} Q^* \Gamma^*, \quad \Omega = \Gamma^{*T} Q^* \phi^*, \quad \Psi = Q_0 + \phi^{*T} Q^* \phi^* \quad (24,25,26)$$

The basic idea of control parameterization is to assume that the value of the i -th control variable can be represented by the product of an approximating function vector $\rho(t)$ and a constant vector y_i , i.e.,

$$u_i(j) = \rho^T(j) y_i \quad \text{for } i = 1, \dots, L; \quad j = 0, \dots, J-1 \quad (27)$$

where

$$\rho(j) = \rho(t_j), \quad \rho^T(t) = [\rho_1(t) \rho_2(t) \dots \rho_M(t)] \quad (28,29)$$

$$y_i = [y_{i1} \ y_{i2} \ \dots \ y_{iM}] \quad (30)$$

In general, $\rho(t)$ can be chosen as a set of orthogonal functions or polynomials. Several $\rho(t)$ will be investigated in the section Simulation Studies.

From equation (27), the control vector $u(t)$ can be represented by

$$u(j) = \rho^*(j) Y \quad \text{for } j = 0, \dots, J-1 \quad (31)$$

where

$$\rho^* = \begin{bmatrix} \rho(j) & 0 & \dots \\ \vdots & \vdots & \vdots \\ \dots & 0 & \rho(j) \end{bmatrix}_{(L \times LM)}, \quad Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_L \end{bmatrix}_{(LM \times 1)} \quad (32)$$

Here, Y represents an unknown vector of control parameters that need to be determined in order to minimize the performance index PI' . Using equations (13) and (31), the control history vector u^* can be expressed as:

$$u^* = \rho^{**} Y \quad (34)$$

where

$$\rho^{**} = \begin{bmatrix} \rho^*(0) \\ \rho^*(1) \\ \vdots \\ \rho^*(J-1) \end{bmatrix}_{(JL \times LM)} \quad (35)$$

Substituting equation (34) into equation (23), PI' can be rewritten as:

$$PI' = \frac{\Delta t}{2} [Y^T \Lambda^* Y + 2 Y^T \Omega^* x_0 + x_0^T \Psi x_0] \quad (36)$$

where

$$\Lambda^* = \rho^{**T} \Lambda \rho^{**}, \quad \Omega^* = \rho^{**T} \Omega \quad (37,38)$$

The necessary condition for optimal Y is

$$\frac{d(PI')}{dY} = 0 \quad (39)$$

which leads to

$$(\Lambda^* + \Lambda^{*T}) Y = -2 \Omega^* x_0 \quad (40)$$

Equation (40) represents a system of linear algebraic equations from which Y can be determined uniquely. In equation (40), only the right-hand side vector depends on the initial condition. As a result, LQ problems involving the same system and the same performance index but different initial conditions can be handled efficiently by using matrix inverse based routines, such as LU decomposition.

A direct application of equation (40) is to choose Y as u^* as suggested in [4] and [5]. Then, the necessary condition for optimality becomes

$$(\Lambda + \Lambda^T) u^* = -2 \Omega x_0 \quad (41)$$

In this case $J = M$, i.e., the number of time intervals is equal to the number of control parameters of each control variable.

As noted above, equation (40) represents a system of linear algebraic equations. Consequently, solving Y from equation (40) involves a computational cost of order $O(L^3 M^3)$ (see, e.g., [8]). Since the computational cost increases cubically with the dimension of Y , it is important to minimize its dimension. A disadvantage of choosing u^* as control parameters is that it makes $J = M$. As a result, solving u^* from equation (41) involves a computational cost of order $O(L^3 J^3)$. In contrast, solving for the control parameter vector Y from equation (40) incurs a computational cost of order $O(L^3 M^3)$. Therefore, by using appropriate approximating functions, M (i.e., the number of control parameters) can be made to be less than J (i.e., the number of time intervals) and, hence, the computational cost can be

significantly reduced while maintaining the same level of accuracy. Further considerations of the computational cost of the control parameterization approach are discussed later. The effectiveness and accuracy of this approach, including the use of different sets of ρ and Y , are addressed via example problems.

LQ Problems with Fixed Terminal States

This subsection considers the optimal control problem defined in equations (1) - (6) but with an additional constraint on the terminal state:

$$\mathbf{x}(t_f) = \mathbf{x}_f \quad (42)$$

From equation (10), this equality constraint can be written as:

$$\mathbf{x}_f = \phi^J \mathbf{x}(0) + \sum_{i=0}^{J-1} \phi^{J-i-1} \mathbf{u}(i) \quad (43)$$

which, equivalently, can be represented as

$$\mathbf{c}\mathbf{u}^* = \mathbf{d} \quad (44)$$

where

$$\mathbf{c} = [\phi^{J-1}\Gamma \quad \phi^{J-2}\Gamma \quad \dots \quad \phi\Gamma \quad \Gamma]_{(N \times JL)}, \quad \mathbf{d} = \mathbf{x}_f - \phi^J \mathbf{x}(0) \quad (45,46)$$

Based on equation (23), the optimization problem can be formulated as:

$$\text{Minimize } PI' = \frac{\Delta t}{2} [\mathbf{u}^T \Lambda \mathbf{u}^* + 2\mathbf{u}^T \Omega \mathbf{x}_0 + \mathbf{x}_0^T \Psi \mathbf{x}_0] \quad (47)$$

subject to $\mathbf{c}\mathbf{u}^* = \mathbf{d}$

From equation (36), with the control parameterization of $\mathbf{u}^* = \rho^{**}\mathbf{Y}$ the problem can be written as

$$\text{Minimize } PI' = \frac{\Delta t}{2} [\mathbf{Y}^T \Lambda^* \mathbf{Y} + 2\mathbf{Y}^T \Omega^* \mathbf{x}_0 + \mathbf{x}_0^T \Psi \mathbf{x}_0] \quad (48)$$

subject to $\mathbf{c}^*\mathbf{Y}^* = \mathbf{d}$

where

$$\mathbf{c}^* = \mathbf{c}\rho^{**} \quad (49)$$

Equations (47) and (48) represent typical formulations of a quadratic programming problem. Solution methods of such problems can be found in the literature, e.g., see [9] and [10].

Minimum Energy Problems

The minimum energy problem is defined as an LQ problem with fixed terminal states, a null state weighting matrix and an identity control weighting matrix (i.e., $Q=0, R=I$). With $Q=0$, the optimization problem defined by equation (48) can be simplified to:

$$\text{Minimize } PI' = \frac{\Delta t}{2} \mathbf{Y}^T \Lambda^* \mathbf{Y} \quad (50)$$

subject to $\mathbf{c}^*\mathbf{Y}^* = \mathbf{d}$

It can be shown that the solution of this problem is

$$\mathbf{Y} = (\Lambda^*)^{-1} (\mathbf{c}^*)^T [(\mathbf{c}^*)^{-1} (\mathbf{c}^*)^T]^{-1} \mathbf{d} \quad (51)$$

The control parameterization approach is particularly efficient in handling such minimum energy problems. By further assuming $R=I$, it can be shown that $\Lambda^* = I$. Applying this result to equation (51), the solution \mathbf{Y} is thus

$$\mathbf{Y} = (\mathbf{c}^*)^T [(\mathbf{c}^*)^{-1} (\mathbf{c}^*)^T]^{-1} \mathbf{d} \quad (52)$$

A similar result is obtained in [4] for discrete-time systems.

DISCUSSION

Computational Considerations

The computational requirements for this control parameterization approach can be classified into three categories. The first part of the computation involves setting up the conditions of optimality (i.e., equation (40)). The associated computational cost of establishing optimality conditions is of order $O(JxL^2xM^2xN^2)$. The second part of the computation relates to the solution of equation (40), a system of equations of order LxM . As a result, the corresponding computational cost is of order $O(L^3xM^3)$. The last part of the computation involves generating the state trajectory from the control parameters. For the proposed approach, this can be done via equation (10). The associated computational cost is of order

$O(sxN^2xL^2)$ where s is the number of points required for the state variables.

Since N and L are given, the computational cost can only be reduced by using smaller M or J . Unfortunately, a smaller M implies that fewer approximating functions are used, potentially sacrificing the accuracy. As shown in equation (17), a smaller J may result in a less accurate approximation of the state variable part of the performance index. However, in general, the accuracy of the control parameterization is determined by the number of control parameters rather than the number of time intervals. As a consequence, J can often be chosen as a small number to reduce the computational cost. It should also be noted that J must be larger than, or at least equal to, M . Otherwise, the control parameters become under-determined and equation (40) becomes singular.

In contrast, solving the Riccati equation by numerical integration requires an order $O(sxcxN^3)$ computation where s is the number of integration steps (and is also the number of points where state and control variables are to be generated), c is an integration routine dependent parameter (for instance, $c=4$ for fourth order Runge-Kutta method) and N is the order of the system. After solving the Riccati equation, an order $O(sxcxN^2)$ method is required to integrate the state equations to generate the response of the state and control variables. This stage is often required to assure the system response meets the prespecified requirements on state and control trajectories.

In searching for the optimal trajectory, usually a rough sketch of the system response is enough to determine if satisfactory results have been achieved. For standard approaches, such as the integration of the state equations, this does not necessarily mean that the number of integration steps (and thus the computational cost) can be reduced since large integration steps produce large numerical errors. This difficulty does not apply to the control parameterization approach since it is assumed that the control variable is single-valued in each time interval regardless of the size of the interval. As a consequence, a small s can often be chosen for the control parameterization approach (but this is not necessarily the case in solving the Riccati equation).

From the above discussions, the control parameterization approach can be computationally more efficient than a Riccati equation solver if (i) J is substantially less than s , and (ii) L is relatively small compared with N . Benchmark results comparing the Riccati-based and control parameterization solutions of LQ problems involving systems of various orders are given below.

A special case is the solution of the minimum energy problem. By choosing \mathbf{u}^* as \mathbf{Y} , it is found that setting up equation (52) incurs a computational cost of order $O(JxN^2)$. Generating the control variables (from the same equation) involves an additional computational cost of order $O(JxLxN + N^3)$. As a result, the computational cost of solving for the optimal control \mathbf{Y} increases only linearly with the number of control parameters. The computational cost of generating the state variable history is of order $O(sxN^2xL^2)$. Compared to the solution of \mathbf{Y} from equation (40) whose computational cost increases cubically with the number of control parameters, the control parameterization approach is particularly efficient in handling minimum energy problems, since the cost increases linearly with J . An example of a minimum energy problem is presented in the following section.

SIMULATION STUDIES

The objective of this section is to evaluate the accuracy and effectiveness of the control parameterization approach by comparing it to a standard Riccati equation solver. For the simulations reported here, both control parameterization and Riccati methods are applied to generate the state and control variables at prespecified equally-spaced points in time for linear optimal control problems with quadratic performance indices.

In the Riccati equation solver, the matrix differential Riccati equation is first integrated backward in time. The results are then used to integrate

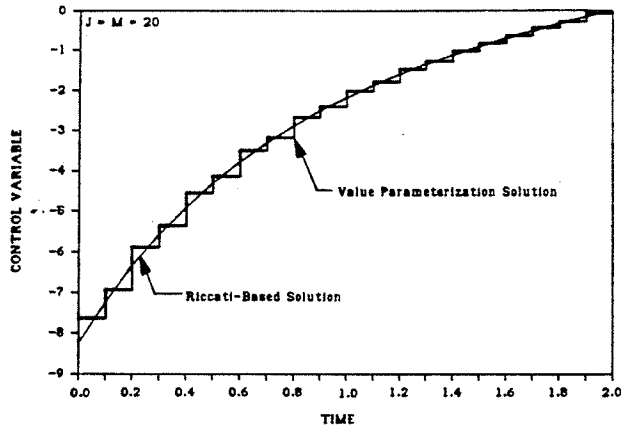


Figure 1a. Control Variable History for Example 1
(Value Parameterization)

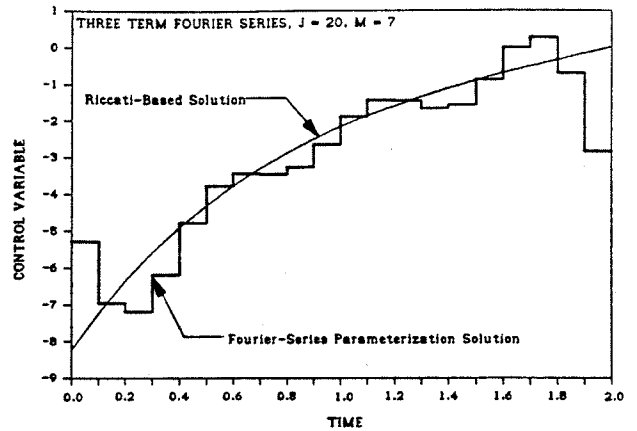


Figure 1b. Control Variable History for Example 1
(Fourier-Series Parameterization)

the state equations (forward in time) to determine the system response. The symmetry of the Riccati equation is utilized to reduce the computational cost. A fourth order Runge-Kutta subroutine is used to integrate numerically both the Riccati and state equations.

In the control parameterization approach, the optimal parameter vector Y is first obtained by solving the system of algebraic equations defined as equation (40). An LU decomposition routine is used in solving these equations. The control variable profile is calculated from equation (34). The state variable profile is computed from equation (11).

The computer programs used in the simulations were written in the "C" language and compiled by a Microsoft Quick C compiler (Version 1.0). Efforts were made to optimize the speed of the computer codes. The simulations were executed on an 8 MHz IBM PC clone with an 8 MHz 8087 coprocessor.

To verify the accuracy of the approach, the value of the performance index from the control parameterization approach was compared to the value from the Riccati equation solver. The time required to execute the program was recorded for each simulation and is used as an index of computational efficiency.

Example 1

Consider a second order, single input system

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ 2 & -1 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u \quad (53)$$

with initial condition $x(0) = [-4 \ 4]^T$ where $x = [x_1 \ x_2]^T$. The weighting matrices of the performance index are

$$H = 0, \quad Q = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}, \quad R = 0.5 \quad (54,55,56)$$

and the terminal time, $t_f = 2.0$.

This example was solved by a Riccati equation solver and by the control parameterization approach with four different approximating functions. These functions are:

$$\rho_1^T = [1 \ 1 \ \dots \ 1]_{1 \times J} \quad (57)$$

$$\rho_2^T = [1 \ \cos \omega_1 t \ \dots \ \cos \omega_K t \ \sin \omega_1 t \ \dots \ \sin \omega_K t]_{1 \times (2K+1)} \quad (58)$$

$$\rho_3^T = [1 \ t \ \cos \omega_1 t \ \dots \ \cos \omega_K t \ \sin \omega_1 t \ \dots \ \sin \omega_K t]_{1 \times (2K+2)} \quad (59)$$

$$\rho_4^T = [1 \ t \ \dots \ t^K]_{1 \times (K+1)} \quad (60)$$

where

$$\omega_k = \frac{2k\pi}{t_f} \quad (61)$$

ρ_1 is the case of using u^* as Y and, for convenience, is called "value parameterization." ρ_2 represents a finite term Fourier series and is called "Fourier-series parameterization." ρ_3 is the case of a finite term Fourier series plus a t term, and is called "Fourier-type parameterization." ρ_4 is the case of a K th order polynomial, and is called "polynomial parameterization."

In this example, the number of time intervals J is chosen to be 20. The control variable histories from the Riccati equation solver and the control parameterization approach using $\rho_1, \rho_2, \rho_3,$ and ρ_4 are plotted in Figures 1a, 1b, 1c, and 1d, respectively. In these figures, the quantized curves refer to the control parameterization solutions and the continuous (smooth) curves refer to the Riccati-based solution. The resulting value of the performance index, the number of control parameters and the percentage error are summarized in Table I for each approach. The percentage error in Table I refers to the error of the value of the performance index with respect to the value of the performance index from the Riccati equation solver.

From Figures 1a to 1d, it is clear that the control parameterization approaches (except the Fourier-series parameterization method) achieve satisfactory results. The inaccuracy of the Fourier-series method is due to the inability of the Fourier series to converge at the boundaries.

The results summarized in Table I show that by using Fourier-type and polynomial parameterization approaches, the number of control parameters can be much less than the number of parameters used in the value parameterization method while maintaining the same level of accuracy. As a result, the computational efficiency is improved without losing significant accuracy. Computational costs of problems of the same and higher order are reported in the next example. It is shown that, by employing Fourier-type and polynomial parameterization, the applicability of the control parameterization approach is greatly improved. On the other hand, the application of the value parameterization approach can only be justified in limited cases due to its computational requirements.

Example 2

The goal of this example is to study the efficiency as well as the accuracy of the control parameterization approaches in handling LQ problems. Here, an N -th order single input system is considered:

$$\dot{x} = \begin{bmatrix} 0 & 1 & & 0 \\ 0 & 0 & 1 & 0 \\ & & & \ddots \\ -1 & -1 & -1 & -1 \end{bmatrix}_{(N \times N)} x + \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix}_{(N \times 1)} u \quad (62)$$

with initial condition $x(0) = [1 \ 1 \ \dots \ 1]^T$.

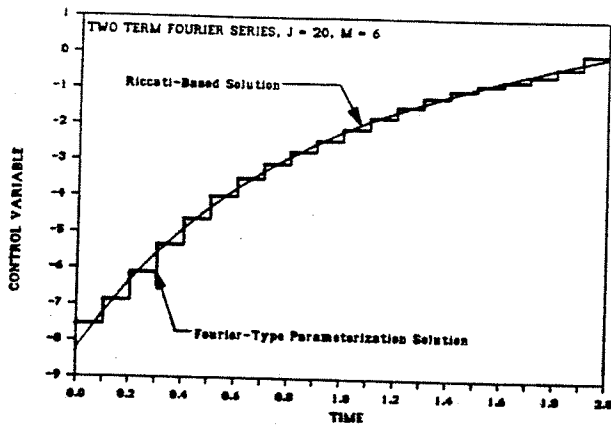


Figure 1c. Control Variable History for Example 1 (Fourier-Type Parameterization)

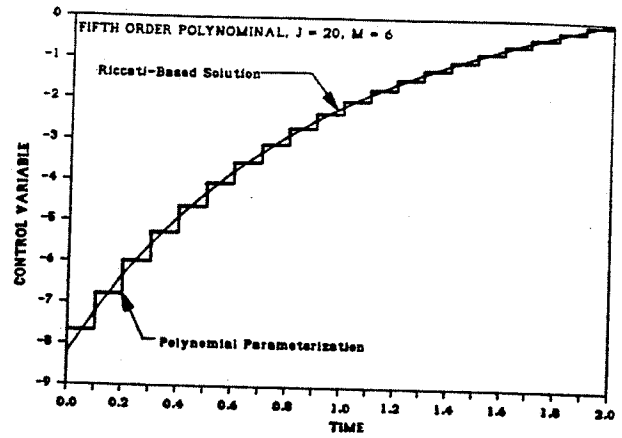


Figure 1d. Control Variable History for Example 1 (Polynomial Parameterization)

The weighting matrices are $Q=I_{(n \times n)}$, $R=1$ and $H=0$. The terminal time $t_f=1$. It is required that $x(t)$ and $u(t)$ be generated at 100 equally spaced points. In this example, the integration time step for the Riccati equation solver is 1/100 time unit. The number of time intervals J is 20. The approximating functions (except for the Fourier-series parameterization which is not used in this example) and the number of control parameters are identical to those used in Example 1. Simulation results for $N=2, \dots, 10$ are summarized in Table II. In Table II the subscript Fourier refers to the Fourier-type parameterization approach instead of the Fourier-series parameterization approach, E refers to error and T refers to computational time.

Table I: Results of Example 1

Approach	Number of Control Parameters M	Performance Index	% Error
Riccati		18.693390	0
Value	20	18.707765	7.69×10^{-2}
Fourier-Series ¹	7	19.171848	2.56
Fourier-Type ²	6	18.709481	8.61×10^{-2}
Polynomial ³	6	18.706141	6.82×10^{-2}

¹Using a three-term Fourier series

²Using a two-term Fourier series

³Using a fifth-order polynomial

Table II shows that the control parameterization approaches used in this example provide satisfactory accuracy in all cases. The value parameterization approach takes less time than solving the Riccati equation when the system order is greater than or equal to six. Table II also shows that, by introducing approximating functions and thus reducing the number of control parameters, the Fourier-type and polynomial parameterization approaches are more efficient than the Riccati equation solver. In addition, as shown in Table II, these two approaches have the same level of accuracy as that of the value parameterization method. From the ratio of the time required for the control parameterization approach and the Riccati equation solver, it is clear that the control parameterization approach is particularly efficient in handling problems whose number of control variables is much less than the order of the system.

Table II: Simulation Results of Example Two

N	PI ¹	%E _V ² $\times 10^{-3}$	%E _F ³ $\times 10^{-2}$	%E _P ⁴ $\times 10^{-3}$	T _R ⁵ sec	T _V /T _R ⁶	T _F /T _R ⁷	T _P /T _R ⁸
2	0.813	8.97	1.30	8.24	5.82	1.756	0.378	0.378
3	2.057	8.36	1.45	7.63	13.9	1.551	0.303	0.303
4	3.818	7.93	1.43	6.76	30.4	1.280	0.249	0.249
5	5.971	7.89	1.41	6.48	57.4	1.096	0.222	0.222
6	8.468	8.11	1.41	6.45	101	0.931	0.201	0.201
7	11.296	8.46	1.44	6.57	168	0.804	0.184	0.184
8	14.454	8.86	1.47	6.76	266	0.695	0.170	0.170
9	17.942	9.23	1.50	6.96	400	0.618	0.160	0.160
10	21.758	9.54	0.54	7.17	583	0.552	0.152	0.152

1. Performance index from Riccati equation solver

2. Percent error of value parameterization approach

3. Percent error of Fourier parameterization approach

4. Percent error of polynomial parameterization approach

5. Computation time of Riccati equation solver

6. Relative computation time of value parameterization approach

7. Relative computation time of Fourier parameterization approach

8. Relative computation time of polynomial parameterization approach

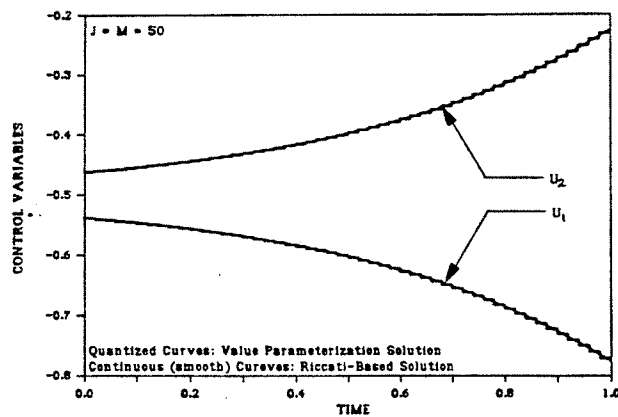


Figure 2. Control Variable History for Example 3
A Minimum Energy Problem

Example 3

This example considers a minimum energy problem described by

$$\dot{x} = \begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix} x + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} u \quad (63)$$

$$PI = \int_0^1 u^T u \, dt \quad (64)$$

The initial condition is $x(0)=[1 \ 0]^T$ and the desired terminal state is $x(1)=[0 \ 0]$. This problem is adapted from [11] (Example 5.3.3) where the closed-form solution is derived. The resulting optimal value of the performance index is 0.26865736. Here, this example is solved using equation (52). The results are summarized in Table III. The results for the case of $J=50$, together with the closed-form solution, are plotted in Figure 2.

An approximate solution has also been obtained using the Riccati equation solver by placing heavy penalty on the terminal states, *i.e.*, H was chosen as $\text{diag}[100,100]$. With the integration step of 1/100 time unit, the value of the performance index was 0.265452 (*i.e.*, an error of 1.2%) and the terminal states were $x_1(1)=0.0076$ and $x_2(1)=0.0023$. The execution time for this program was 6.4 second. The accuracy of this Riccati-based approach can be improved by using larger elements for H . However, this requires a smaller integration step size to avoid numerical instability problems, increasing the computational cost.

In contrast, as shown in Table III, the control parameterization approach requires less computational time and also provides better accuracy than the Riccati equation solver. The requirements on the terminal states are also satisfied. Furthermore, Table III shows that the computational cost, as expected, grows linearly with the number of the control parameters which is equal to the number of time intervals in this case.

Table III: Minimum Energy Problem of Example 3

J	PI	% Error	T(sec)
50	0.26865985	9.3×10^{-4}	0.7
100	0.26855798	2.3×10^{-4}	1.2
150	0.26865764	1.0×10^{-4}	1.7
200	0.26865752	6.0×10^{-5}	2.2

CONCLUSION

Based on the technique of mathematical programming, this paper develops a control parameterization approach for LQ problems which are unconstrained or have fixed terminal states. The approach converts an LQ problem into a quadratic programming problem. For an unconstrained LQ problem, the necessary and sufficient condition for optimality is derived as a system of linear algebraic equations which is readily solved.

As demonstrated by simulation results, the control parameterization approach achieves high accuracy and is computationally faster than a Riccati equation solver in handling problems whose number of control variables is small relative to the order of the system. A major advantage of this approach is that it can handle minimum energy problems with considerable efficiency.

REFERENCES

- [1] Tabak, D., "Applications of Mathematical Programming Techniques in Optimal Control: A Survey," *IEEE Transaction on Automatic Control*, Vol. AC-15, December, pp. 688-690, 1970.
- [2] Kraft, D., "Comparing Mathematical Programming Algorithms Based on Lagrangian Functions for Solving Optimal Control Problems," *Control Applications of Nonlinear Programming*, ed. H.E. Rauch, Pergamon Press, Oxford, 1980.
- [3] Goh, C.J. and Teo, K.L., "Control Parameterization: A Unified Approach to Optimal Control Problems with General Constraints," *Automatica*, Vol. 24, No. 1, pp. 3-18, 1988.
- [4] Canon, M.D., Cullum, C.D. and Polak, E., *Theory of Optimal Control and Mathematical Programming*, McGraw-Hill, Inc., New York, NY, 1970.
- [5] Tabak, D. and Kuo, B.C., *Optimal Control by Mathematical Programming*, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1971.
- [6] Ghonaimy, M.A.R. and El-Zorkani, H.I., "A Direct Method for Computing Optimal Feedback Control for Linear System," *International Journal of Control*, Vol. 16, No. 4, pp. 751-762, 1972.
- [7] Yen, V. and Nagurka, M.L., "Fourier-Based State Parameterization for Linear Quadratic Optimal Control," Technical Paper 88-WA/DSC-7, ASME Winter Annual Meeting, Chicago, IL, November 27 - December 2, 1988.
- [8] Westlake, J.R., *A Handbook of Numerical Matrix Inversion and Solution of Linear Equations*, John Wiley & Sons, Inc., New York, NY.
- [9] Wolfe, P., 1959, "The Simplex Method for Quadratic Programming," *Econometrica*, Vol. 27, No. 3, July, pp. 382-398, 1968.
- [10] Van De Panne, C. and Whinston, A., "The Simplex and the Dual Method for Quadratic Programming," *Operation Research Quarterly*, Vol. 15, December, pp. 355-388, 1964.
- [11] Owen, D.H., *Multivariable and Optimal Systems*, Academic Press, London, 1981.