INT. J. SYSTEMS SCI., 1992, VOL. 23, NO. 12, 2125–2139

Development of linear quadratic control laws via control parametrization

MARK L. NAGURKA[†] and VINCENT YEN[‡]

A control parametrization approach for determining the near optimal solution of linear quadratic (LQ) problems is developed. By assuming each control variable to be piecewise continuous, the proposed approach converts an LQ problem into an unconstrained quadratic programming problem. The near optimal control response can then be determined by solving a system of linear algebraic equations. The control parametrization approach can eliminate the major repetitive computations in designing an LQ control law, a process that often involves multiple adjustments of different weighting matrices. This feature makes the proposed approach a computationally attractive tool for LQ controller design. Simulation studies show that the control parametrization approach is particularly well suited for large scale systems that possess a small control/state dimension ratio.

1. Introduction

The idea of converting optimal control problems into mathematical programming (MP) problems via trajectory parametrization is not new. For example, Hicks and Ray (1971) and Goh and Teo (1988) proposed casting optimal control problems as MP problems using control parametrization methods. Sirisena and Chou (1981) and Nagurka and Yen (1990) suggested using state parametrization approaches for converting optimal control problems into MP problems. Neuman and Sen (1973) and Vlassenbroeck and Van Dooren (1988) parametrized both state and control variables to convert optimal control problems into MP problems.

The advantages of using trajectory parametrization are most evident in solving non-linear and/or constrained optimal control problems. The solutions of these optimal control problems are often impractical to obtain using traditional methods such as variational techniques. In general, the difficulties can be overcome by drawing upon the power of non-linear programming solution algorithms. As a result, most previous trajectory parametrization approaches have been developed for non-linear and/or constrained optimal control problems.

This paper concentrates on the application of a control parametrization approach for solving the unconstrained linear quadratic (LQ) problem. The study of such a fundamental problem is valuable for several reasons. First, the LQ problem represents one of the most important problems of linear optimal control, and its solution serves as the basis for linear feedback system design. Second, by using a technique such as quasi-linearization, an unconstrained non-linear optimal control problem can be converted into a sequence of LQ problems (e.g., see Bashein and Enns 1972). An efficient solution method for LQ problems can thus reduce the computational cost in solving unconstrained non-linear optimal control problems.

Received 23 March 1992.

[†] Department of Mechanical Engineering, Carnegie Mellon University, Pittsburgh, Pennsylvania, U.S.A.

[‡] Department of Mechanical Engineering, National Sun Yat-Sen University, Kaohsiung, Taiwân, Republic of China.

Third, unlike non-linear and/or constrained LQ problems, traditional methods for solving unconstrained LQ problems are very well developed. It is thus interesting to see whether trajectory parametrization holds any advantage over traditional methods in solving unconstrained LQ problems.

This paper is organized as follows. The basic methodology of converting an LQ problem into an unconstrained quadratic programming problem is described in § 2; § 3 illustrates the application of the proposed approach for solving LQ problems with varying performance index weightings; the advantages and drawbacks of the proposed approach are addressed in § 4; conclusions are drawn in § 5.

2. Basic methodology

The behaviour of a linear, time-invariant dynamical system is governed by the state-space model

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \tag{1}$$

with initial condition

$$\mathbf{x}(0) = \mathbf{x}_0 \tag{2}$$

where $\mathbf{x}$ is an $N \times 1$ state vector, $\mathbf{u}$ is an $L \times 1$ control vector, $\mathbf{A}$ is an $N \times N$ system matrix, and $\mathbf{B}$ is an $N \times L$ control influence matrix. The design goal is to find the optimal control $\mathbf{u}(t)$ and the corresponding state $\mathbf{x}(t)$ in the interval $[0, \tau]$ that minimizes the quadratic performance index

$$\mathrm{PI} = \mathbf{x}^{\mathrm{T}}(\tau)\mathbf{H}\mathbf{x}(\tau) + \int_0^{\tau} [\mathbf{x}^{\mathrm{T}}\mathbf{Q}\mathbf{x} + \mathbf{u}^{\mathrm{T}}\mathbf{R}\mathbf{u}]\,\mathrm{d}t \tag{3}$$

Here, $\mathbf{H}$ and $\mathbf{Q}$ are real, symmetric, positive-semidefinite matrices and $\mathbf{R}$ is a real, symmetric, positive-definite matrix. It is assumed that the state and control vectors are not bounded, the terminal time τ is fixed, and the terminal state $\mathbf{x}(\tau)$ is free.

The first step of the control parametrization approach is to divide the interval $[0, \tau]$ into J equal intervals $[t_0, t_1], [t_1, t_2], \ldots, [t_{J-1}, t_J]$, where $t_0 = 0$ and $t_J = \tau$. It is assumed that each of the L control variables has a single value in each interval $[t_{j-1}, t_j]$, $j = 1, \ldots, J$, and is piecewise continuous in $[0, \tau]$. The system can thus be converted into the following discrete-time model:

$$\mathbf{x}(j+1) = \phi\mathbf{x}(j) + \gamma\mathbf{u}(j) \quad \text{for } j = 0, \ldots, J-1 \tag{4}$$

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

$$\phi = \exp(\mathbf{A}\,\Delta t), \quad \gamma = \int_0^{\Delta t} \exp(\mathbf{A}t)\,\mathrm{d}t\,\mathbf{B} \tag{5\,a, b}$$

Here, the interval length is written as Δt and, for the sake of simplicity, $\mathbf{x}(t_j)$ and $\mathbf{u}(t_j)$ are denoted by $\mathbf{x}(j)$ and $\mathbf{u}(j)$, respectively. Multiple applications of (4) lead to the following equation:

$$\mathbf{x}(j) = \phi^j\mathbf{x}(0) + \sum_{i=0}^{j-1} \phi^{j-i-1}\gamma\mathbf{u}(i) \tag{6}$$

Based on (6), the relationship between the state and control trajectories can be established as

$$\mathbf{X} = \mathbf{\Phi}\mathbf{x}(0) + \mathbf{\Gamma}\mathbf{U} \tag{7}$$

where

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}(1) \\ \mathbf{x}(2) \\ \vdots \\ \mathbf{x}(J) \end{bmatrix}_{JN\times 1}, \quad \mathbf{U} = \begin{bmatrix} \mathbf{u}(0) \\ \mathbf{u}(1) \\ \vdots \\ \mathbf{u}(J-1) \end{bmatrix}_{JL\times 1} \tag{8 a, b}$$

$$\mathbf{\Phi} = \begin{bmatrix} \phi \\ \phi^2 \\ \vdots \\ \phi^J \end{bmatrix}_{JN\times N}, \quad \mathbf{\Gamma} = \begin{bmatrix} \gamma & \mathbf{0} & & \dots & & \mathbf{0} \\ \phi\gamma & \gamma & \mathbf{0} & \dots & \dots & \mathbf{0} \\ \vdots & \vdots & & \ddots & & \\ \phi^{J-1}\gamma & \phi^{J-2}\gamma & & \dots & & \gamma \end{bmatrix}_{JN\times JL} \tag{9 a, b}$$

The adjoining subscripts represent the dimensions of the matrices. By linking the control and the state, (7) can be used to convert the performance index into a function of control parameter vector $\mathbf{U}$.

The control integral part of the performance index can be rewritten as

$$\int_0^T [\mathbf{u}^{\mathrm{T}}\mathbf{R}\mathbf{u}]\,dt = \sum_{j=0}^{J} \mathbf{u}^{\mathrm{T}}(j)\mathbf{R}\mathbf{u}(j)\,\Delta t \tag{10}$$

Note that due to the assumed unit-step nature of the control variables, (10) is not an approximation. By employing Simpson's rule, the terminal state and state integral part of the performance index can be approximated by

$$\mathbf{x}^{\mathrm{T}}(J)\mathbf{H}\mathbf{x}(J) + \int_0^T [\mathbf{x}^{\mathrm{T}}\mathbf{Q}\mathbf{x}]\,dt \approx \sum_{j=0}^{J} \mathbf{x}^{\mathrm{T}}(j)\mathbf{Q}_j\mathbf{x}(j)\Delta t \tag{11}$$

where the matrices $\mathbf{Q}_j$ are defined as

$$\mathbf{Q}_j = \begin{cases} \frac{1}{3}\mathbf{Q}, & \text{for } j = 0 \\ \frac{2}{3}\mathbf{Q}, & \text{for } j = 2, 4, \dots, J-2 \\ \frac{4}{3}\mathbf{Q}, & \text{for } j = 1, 3, \dots, J-1 \\ \frac{1}{3}\mathbf{Q} + \mathbf{H}, & \text{for } j = J \end{cases} \tag{12}$$

with J assumed to be an even number. Note that the exact value of this part of the performance index can be computed as a finite sum of the state vector values (e.g. Dorato and Levis 1971). Simpson's rule is adopted due to its satisfactory accuracy and simplicity.

By combining (10) and (11), the performance index PI can now be approximated by PI*:

$$\mathrm{PI}^* = \frac{\Delta t}{2}(\mathbf{U}^{\mathrm{T}}\mathbf{R}^*\mathbf{U} + \mathbf{X}^{\mathrm{T}}\mathbf{Q}^*\mathbf{X} + \mathbf{x}^{\mathrm{T}}(0)\mathbf{Q}_0\mathbf{x}(0)) \tag{13}$$

where

$$\mathbf{R}^* = \begin{bmatrix} \mathbf{R} & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \mathbf{R} \end{bmatrix}_{JL\times JL}, \quad \mathbf{Q}^* = \begin{bmatrix} \mathbf{Q}_1 & & & \mathbf{0} \\ & \mathbf{Q}_2 & & \\ & & \ddots & \\ \mathbf{0} & & & \mathbf{Q}_J \end{bmatrix}_{JN\times JN} \tag{14 a, b}$$

By substituting (7) into (13), the converted performance index can be rewritten as

$$\mathrm{PI}^* = \frac{\Delta t}{2}(\mathbf{U}^{\mathrm{T}}\mathbf{\Lambda}\mathbf{U} + \mathbf{U}^{\mathrm{T}}\mathbf{\Omega}\mathbf{x}_0 + \mathbf{x}_0^{\mathrm{T}}\mathbf{\Psi}\mathbf{x}_0) \tag{15}$$

where

$$\mathbf{\Lambda} = \mathbf{R}^* + \mathbf{\Gamma}^{\mathrm{T}}\mathbf{Q}^*\mathbf{\Gamma}, \quad \mathbf{\Omega} = \mathbf{\Gamma}^{\mathrm{T}}\mathbf{Q}^*\mathbf{\Phi}, \quad \mathbf{\Psi} = \mathbf{Q}_0 + \mathbf{\Phi}^{\mathrm{T}}\mathbf{Q}^*\mathbf{\Phi} \tag{16a–c}$$

The necessary condition for minimum PI* is

$$\frac{d(\mathrm{PI}^*)}{d\mathbf{U}} = \mathbf{0} \tag{17}$$

which leads to

$$(\mathbf{\Lambda} + \mathbf{\Lambda}^{\mathrm{T}})\mathbf{U} = 2\mathbf{\Lambda}\mathbf{U} = -\mathbf{\Omega}\mathbf{x}_0 \tag{18}$$

Equation (18) represents a system of linear algebraic equations from which **U** can be determined. In (18), 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.

An important characteristic of (18) is that its dimension ($L \times J$) is not a function of the system order. As a result, the computational cost of solving (18) is proportional to $O(L^3J^3)$ and is independent of the number of state variables. Consequently, this control parametrization approach can efficiently handle high order LQ problems that have a few control variables provided that the number of time intervals J is not too large. The following example investigates the efficiency and accuracy of the proposed approach, with special attention given to the influence of the number of control variables.

Example 1

Consider a class of 10th order systems defined as

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 1 & 0 & \dots & \\ 0 & 0 & 1 & 0 & \dots \\ \vdots & \vdots & & \ddots & \\ 0 & 0 & & & 1 \\ -1 & -1 & \dots & -1 & -1 \end{bmatrix}_{10\times 10} \mathbf{x} + \mathbf{B}_{10\times L}\mathbf{u}, \quad \text{for } L = 1, \dots, 10 \tag{19}$$

with initial condition

$$\mathbf{x}(0) = [1 \quad 1 \quad \dots \quad 1]^{\mathrm{T}} \tag{20}$$

Note that the number of control variables L of these systems ranges from 1 to 10. The corresponding control influence matrix **B** has dimension $10 \times L$ and contains the first L columns of a 10×10 identity matrix. The objective is to generate the optimal state and control vectors at $t = 0{\cdot}1, 0{\cdot}2, \dots, 1{\cdot}0$ with the performance index specified as

$$\mathrm{PI} = \frac{1}{2}\int_0^1 [\mathbf{x}^{\mathrm{T}}\mathbf{x} + \mathbf{u}^{\mathrm{T}}\mathbf{u}]\, dt \tag{21}$$

These problems were solved using the control parametrization approach and a transition matrix approach. The basic idea of the transition matrix approach is to derive the necessary condition of optimality of an LQ problem as a linear two-point boundary-value problem (TPBVP). By evaluating the transition matrix of the corresponding system matrix, the TPBVP is converted into a standard initial value problem that can be readily solved. For more information on the transition matrix method, readers are referred to Speyer (1986).

The computer programs used in this and the next example were written in the C language and compiled with a Turbo C compiler (Version 2.0). Efforts were made to optimize the speed of the computer codes. The simulations were executed on a 25 MHz 386 PC with an 80387 coprocessor. To verify the accuracy of the approach, the value of the performance index from the control parametrization approach was compared to the value obtained from the transition matrix method. The execution time (in seconds) was recorded for each simulation and is used as an index of computational efficiency. The results are summarized in Table 1.

The control variable histories from the transition matrix method and the control parametrization approach for the case of $L = 1$ are plotted in Fig. 1. In this figure, the quantized curve refers to the control parametrization solution. The continuous (smooth) curve refers to the transition matrix solution. Figure 1 shows that the control response generated from the control parametrization approach represents a good approximation of the true optimal solution obtained by the transition matrix method.

Several conclusions can be drawn from the results of Table 1. First, the inaccuracy of the control parametrization approach increases with the number of control variables, although it provides satisfactory engineering accuracy (less than 1% error in performance index value) in all cases. Second, the computational cost of the transition matrix approach seems to be insensitive to the number of control variables, whereas the execution time of the control parametrization approach increases as the number of control variables grows. In particular, the execution time of the control parametrization approach is reduced in solving problems where the

	Transition matrix		Control parametrization		Comparison	
L	PI	Time	PI	Time	% Error†	% Time‡
1	21·6956	5·8	21·6983	2·3	0·01	39·66
2	19·6023	5·8	19·6089	2·7	0·03	46·55
3	17·5887	5·8	17·5989	3·2	0·06	55·17
4	15·7297	5·8	15·7427	4·0	0·08	68·97
5	14·0128	5·8	14·0282	4·9	0·11	84·48
6	12·4330	5·8	12·4503	6·1	0·14	105·17
7	11·0336	5·9	11·0525	7·6	0·17	128·81
8	10·0080	5·9	10·0281	9·3	0·20	157·63
9	9·6834	6·0	9·7061	11·4	0·23	190·00
10	9·3877	6·0	9·4103	14·3	0·24	238·33

† Percentage difference of performance index of control parametrization approach relative to performance index value of transition-matrix approach.

‡ Percentage of execution time of control parametrization approach relative to execution time of transition-matrix approach.

Table 1. Summary of simulation results of Example 1.

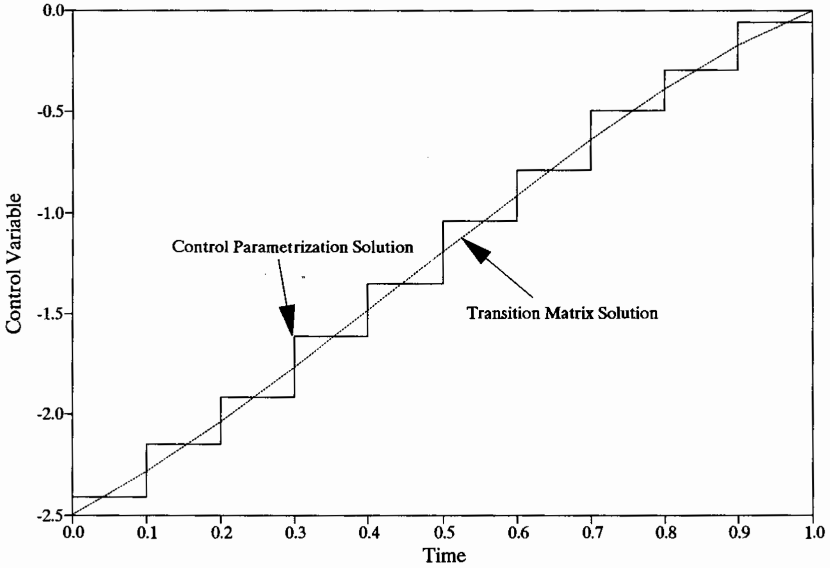

Figure 1. Control variable history for Example 1.

number of control variables is less than or equal to half the number of state variables. This result is expected since, as mentioned earlier, the computational cost for solving (18) decreases with the number of control variables. This characteristic makes the control parametrization approach a very attractive tool for handling 'real-world' high order LQ problems. In many practical large scale engineering systems, the dimension of the control vector is only a small fraction of the dimension of the state vector.

As shown in Table 1, the control parametrization approach becomes less efficient as the dimension of the control vector approaches the system order. This may suggest that the proposed approach is best suited for systems whose control/state dimension ratio is small. In the following section, additional features of the control parametrization approach are explored. It is shown that the control parametrization approach can be used for efficient design of LQ control laws for systems with any control/state dimension ratio.

3. Designing LQ control laws with different weighting matrices

In the synthesis of an LQ controller, the elements of the weighting matrices need to be determined. Unfortunately, general methods for selecting $\mathbf{Q}$ and $\mathbf{R}$ do not seem to exist. As a consequence, weighting matrices are typically determined by trial-and-error methods, generally requiring many iterations. In practice, computationally efficient 'open-loop' approaches such as the transition matrix method are often used in this iterative design process. For example, a computer simulation of an open-loop method is conducted to determine the state and control histories for different test sets of $\mathbf{Q}$ and $\mathbf{R}$. The state and control trajectories are then inspected by design engineers and this process is repeated until weighting matrices are identified that give satisfactory performance (e.g. speed of response) without saturation of the control variables. Once the desired $\mathbf{Q}$ and $\mathbf{R}$ have been found, a computationally more intensive 'closed-loop' approach such as a Riccati equation-based method is used for implementation. In summary, an open-loop approach such as a transition matrix method is often employed in the early stages of LQ controller design. Given a satisfactory design, a closed-loop approach is then implemented.

In this section, the possibility of reducing the computational cost of the control parametrization approach is investigated. By eliminating the repetitive part of the computations, the control parametrization approach can achieve higher efficiency in the iterative process of LQ control law development. In the following subsection, modifications of the state weighting matrix are investigated.

3.1. *Change of state weighting matrix*

It is assumed that the LQ problem defined in § 1 has been solved at least once for a given set of $\mathbf{Q}$ and $\mathbf{R}$. The goal of this subsection is to develop a method to accelerate the generation of new near optimal state and control trajectories when $\mathbf{Q}$ is subsequently modified.

The discrete time model defined by (4) is not a function of the weighting matrices. Equations (5)–(9) are also independent of the weighting matrices. Consequently, the results of these equations, once obtained, need not be recomputed if changes are made only to the weighting matrices. The next step is to analyse the structure of the approximating performance index PI* defined by (15). The purpose is to eliminate redundant computations associated with the construction of the approximate performance index. It can be shown that (15) can be decomposed into the following form:

$$\mathrm{PI}^* = \mathrm{PI}_1^* + \mathrm{PI}_2^* + \mathrm{PI}_3^* \tag{22}$$

where

$$\mathrm{PI}_1^* = \frac{1}{2}\sum_{j=0}^{J=1} \mathbf{x}^{\mathrm{T}}(0)(\phi^{\mathrm{T}})^j \mathbf{Q}_j \phi^j \mathbf{x}(0) \tag{23}$$

$$\mathrm{PI}_2^* = \frac{1}{2}\sum_{j=0}^{J-1}\sum_{i=0}^{j-1} \mathbf{u}^{\mathrm{T}}(i)\gamma^{\mathrm{T}}(\phi^{\mathrm{T}})^{j-i-1} \mathbf{Q}_j \phi^j \mathbf{x}(0) \tag{24}$$

$$\mathrm{PI}_3^* = \frac{1}{2}\sum_{j=0}^{J-1}\sum_{i=0}^{j-1}\sum_{k=0}^{j-1} \mathbf{u}^{\mathrm{T}}(i)\gamma^{\mathrm{T}}(\phi^{\mathrm{T}})^{j-i-1} \mathbf{Q}_j \phi^{j-k-1}\gamma \mathbf{u}(k) \tag{25}$$

In (23), the square matrix associated with the quadratic term of the initial state vector $\mathbf{x}(0)$ has the following property:

$$\operatorname{vec}((\phi^{\mathrm{T}})^j \mathbf{Q}_j \phi^j) = ((\phi^{\mathrm{T}})^j \otimes (\phi^{\mathrm{T}})^j) \operatorname{vec}(\mathbf{Q}_j) \tag{26}$$

where the vector valued function vec and the Kronecker product $\otimes$ are defined, respectively, as

$$\operatorname{vec}(\mathbf{C}) = \begin{bmatrix} \mathbf{c}_1 \\ \mathbf{c}_2 \\ \vdots \\ \mathbf{c}_n \end{bmatrix}, \quad \mathbf{D} \otimes \mathbf{E} = \begin{bmatrix} d_{11}\mathbf{E} & d_{12}\mathbf{E} & \dots & d_{1q}\mathbf{E} \\ d_{21}\mathbf{E} & & & \\ \vdots & & & \vdots \\ d_{p1}\mathbf{E} & \dots & & d_{pq}\mathbf{E} \end{bmatrix} \tag{27 a, b}$$

In (27 a), n is the row number of matrix $\mathbf{C}$ and $\mathbf{c}_i$ is the ith column vector of matrix $\mathbf{C}$ for $i = 1, \dots, n$. In (27 b), matrix $\mathbf{D}$ has dimension $p \times q$. For further details on the properties of the Kronecker product, readers are referred to Brewer (1978).

The advantage of reformulating (23) as (26) is that it decouples the state weighting matrix from the system parameters. In particular, the term associated with transition matrix ϕ can be stored in computer memory and retrieved later when the state weighting matrix $\mathbf{Q}$ is changed. The number of variables that need to be stored is JN^4. The same technique can be applied to (24) and (25) to further reduce the repetitive parts of the computations. In particular, (24) and (25) can be reformulated as

$$\operatorname{vec}(\gamma^{\mathrm{T}}(\phi^{\mathrm{T}})^{j-i-1}\mathbf{Q}_j\phi^j) = ((\phi^{\mathrm{T}})^j \otimes [\gamma^{\mathrm{T}}(\phi^{\mathrm{T}})^{j-i-1}])\operatorname{vec}(\mathbf{Q}_j) \tag{28}$$

and

$$\operatorname{vec}(\gamma^{\mathrm{T}}(\phi^{\mathrm{T}})^{j-i-1}\mathbf{Q}_j(\phi^j)^{j-k-1}\gamma) = ([\gamma^{\mathrm{T}}(\phi^{\mathrm{T}})^{j-k-1}] \otimes [\gamma^{\mathrm{T}}(\phi^{\mathrm{T}})^{j-i-1}])\operatorname{vec}(\mathbf{Q}_j) \tag{29}$$

From (28) and (29), the number of variables that need to be stored in computer memory is LJ^2N^3 and $J^2L^2N^2$, respectively. Hence, the memory requirement for such a method is quite large and the improvement in computational speed can only be achieved when sufficient storage is available.

3.2. *Change of control weighting matrix*

This subsection considers the situation in which only the control weighting matrix $\mathbf{R}$ is changed. In this case, the performance index (15) and the necessary condition of optimality (18) can be updated easily provided that the state weighting part of the performance index in (16) has been stored. The number of variables that need to be retained in memory is $L^2J^2 + LJN + N^2$. The computer storage requirement is much less than the requirement for the technique developed in § 3.1.

After the updating process, the necessary condition of optimality must be solved again to regenerate the optimal solution. The following part of this subsection investigates the possibility of circumventing the complete repetition of the process of solving (18). In particular, LDL matrix factorization and a matrix inverse updating technique are suggested.

Attention is first given to LDL factorization. It can be shown that the matrix $\mathbf{\Lambda}$, defined by (16*a*), is a positive-definite matrix and thus can be written as

$$2\mathbf{\Lambda} = \mathbf{LDL}^{\mathrm{T}} \tag{30}$$

where $\mathbf{L}$ is a lower-triangular matrix with unit diagonals and $\mathbf{D}$ is a diagonal matrix. The computational requirement of this LDL factorization process is of order $O(n^3/3)$ (see e.g. Golub and Van Loan 1989) where n is the dimension of matrix $\mathbf{\Lambda}$. In comparison, solving $\mathbf{U}$ from the factorized equation (18) requires only $O(n^2)$ floating point steps. Therefore, the LDL factorization process represents the predominant part of the computational cost for solving a system of linear algebraic equations. Note that LDL factorization is essentially a matrix inverse scheme. This is because a system of linear algebraic equations can also be solved in $O(n^2)$ floating steps when its matrix inverse is known.

The LDL matrix factorization method has two distinct advantages in our application. First, it can solve the necessary condition of optimality more efficiently than other standard linear algebraic equation solvers. Second, the LDL factorization provides an efficient means for updating factorized matrix $\mathbf{L}$ and $\mathbf{D}$ when the matrix of interest is modified via a single or a series of rank-one changes. This LDL factorization-based matrix inverse updating technique and its application to the proposed optimal control approach are addressed next.

Suppose that (18) has been factorized and is now of the following form:

$$2\mathbf{\Lambda U} = (\mathbf{LDL}^{\mathrm{T}})\mathbf{U} = -\mathbf{\Omega x}_0 \tag{31}$$

It is further assumed that a rank-one change is applied to matrix $\mathbf{\Lambda}$ so that the new matrix $\mathbf{\Lambda}^*$ can be represented as

$$2\mathbf{\Lambda}^* = 2\mathbf{\Lambda} + \alpha\mathbf{yy}^{\mathrm{T}} = \mathbf{L}^*\mathbf{D}^*\mathbf{L}^{*\mathrm{T}} \tag{32}$$

where $\mathbf{y}$ is a column vector and α is a real scalar variable. It is assumed that the modified matrix $\mathbf{\Lambda}^*$ remains positive-definite. For any given $\mathbf{y}$, it can be shown that there exists a column vector $\mathbf{g}$ that satisfies the following equation:

$$\mathbf{Lg} = \mathbf{y} \tag{33}$$

Substituting (33) into (32) gives

$$2\mathbf{\Lambda}^* = \mathbf{L}(\mathbf{D} + \alpha\mathbf{gg}^{\mathrm{T}})\mathbf{L}^{\mathrm{T}} \tag{34}$$

The middle term of (34) can be factorized as

$$\mathbf{D} + \alpha\mathbf{gg}^{\mathrm{T}} = \hat{\mathbf{L}}\hat{\mathbf{D}}\hat{\mathbf{L}}^{\mathrm{T}} \tag{35}$$

From (32) and (35), the following results can be obtained

$$\mathbf{L}^* = \mathbf{L}\hat{\mathbf{L}}, \quad \mathbf{D}^* = \hat{\mathbf{D}} \tag{36a,b}$$

Note that such an updating operation has the same effect as updating the matrix inverse. Due to the special structure of LDL factorization, the new $\mathbf{L}^*$ and $\mathbf{D}^*$ of $\mathbf{\Lambda}^*$ can thus be generated with only a computational cost of order $O(n^2)$. The computational requirement can be further reduced when the leading rows of $\mathbf{y}$ are zeros. The detailed algorithm for this LDL factorization-based matrix inverse updating scheme can be found in Gill *et al.* (1974) and Gill and Murray (1977). The next question is how to apply this matrix inverse updating technique to the solution of the LQ problem.

Assuming that the control weighting matrix is diagonal and is changed from $\mathbf{R}$ into $\mathbf{P}$, then matrix $\mathbf{\Lambda}$ of (18) can be modified using a series of rank-one changes. Specifically, the modification can be represented as

$$\mathbf{\Lambda}^* = \mathbf{\Lambda} + \sum_{i=1}^{L}\sum_{j=1}^{J}(p_i - r_i)\mathbf{y}_{ij}^{\mathrm{T}}\mathbf{y}_{ij} \tag{37}$$

where r_i and p_i are the ith diagonal elements of matrices $\mathbf{R}$ and $\mathbf{P}$, respectively. The term $\mathbf{y}_{ij}$ is a column vector with the only non-zero element being unity in row $(i-1)J+j$.

In designing an LQ controller, the weighting on each of the control variables is frequently scaled one at a time. If this is the case, the above transformation can be simplified to

$$\mathbf{\Lambda}^* = \mathbf{\Lambda} + \sum_{j=1}^{J}(p_i - r_i)\mathbf{y}_{ij}^{\mathrm{T}}\mathbf{y}_{ij} \tag{38}$$

Here, it is assumed that only the ith diagonal term of $\mathbf{R}$ is changed. The following example will address the comparative efficiency of the proposed LDL factorization-based updating method in solving LQ problems with varying control weighting.

Example 2

The specifications of the system and performance index used in this example are identical with those of Example 1. The emphasis here is to study the relative computational efficiency in solving LQ problems with varying control weighting.

Three control parametrization strategies were tested. In the first approach, the general control weighting updating method is employed to eliminate the computations associated with (4)–(14). This method does not use the LDL-based matrix inverse updating technique. As a result, the complete process for solving (18) has to be repeated. The advantage of this approach is that the control weighting **R** can be changed in an arbitrary manner provided that it remains positive-definite. The second and third techniques are applicable only when changes are made to the diagonal terms of **R**. In the second case, the diagonal control weighting updating method permits changes to all or part of the diagonal terms of **R**. The necessary condition of optimality can be modified using (37). In the last method, the single control weighting updating method, only one control variable weighting is changed at a time. Consequently, (38) is used to update the necessary condition of optimality. Additionally, the last two methods use the matrix inverse updating method described above to speed up the solution of (18).

The benchmark results are summarized in Table 2. The results show that the computational cost of the three control parametrization strategies increases with the number of control variables. However, the optimal solutions are generated more slowly via the transition matrix method when changes are made to the control weighting matrix. Only in the case of the general control weighting updating method (i.e. the first method) with $L = 10$ is the execution time faster for the transition matrix method.

In comparing the three strategies, the results show that the general control

		Control weighting updating methods of control parametrization approach		
L	Transition matrix	General†	Diagonal‡	Single§
1	5·8	0·04	0·04	0·04
2	5·8	0·12	0·13	0·11
3	5·8	0·29	0·28	0·21
4	5·8	0·56	0·54	0·33
5	5·8	1·00	0·95	0·51
6	5·8	1·60	1·49	0·71
7	5·9	2·45	2·24	0·95
8	5·9	3·52	3·16	1·21
9	6·0	4·89	4·37	1·54
10	6·0	6·85	5·84	1·90

† Execution time of control parametrization approach when control weighting matrix is arbitrarily changed.

‡ Execution time of control parametrization approach when all or part of the diagonal terms of control weighting matrix are changed.

§ Execution time of control parametrization approach when only one diagonal term of control weighting matrix is changed.

Table 2. Summary of execution time of Example 2.

weighting updating method is the slowest. The computational efficiency can be improved by employing the diagonal control weighting method if changes are made only to the diagonal terms. The single control weighting updating method, involving changes to only one of the diagonal terms of **R**, is the fastest. Table 2 shows that for the case of $L = 1$, the three control parametrization strategies run two orders of magnitude faster than the transition matrix approach. Moreover, this order of saving in computational time is maintained up to $L = 5$ for the single control weighting updating method. These results suggest a particularly efficient strategy for handling problems whose number of control variables is significantly less than the system order.

In this and the previous example, the final time τ and the number of time intervals J have been assumed to be fixed. However, it may be necessary to increase J as τ grows larger in order to maintain the accuracy of the control parametrization approach. In the following example, the influence of final time τ is investigated.

Example 3

The system and performance index in this example are the same as those of Example 1 with the following exceptions: (i) the final time τ is no longer fixed, and (ii) only the case of $L = 1$ is considered. The computer programs used in this example were written in the C language and compiled with a Turbo C++ compiler. The simulations were executed on a 25 MHz 486 PC.

To investigate the influence of final time, problems with $\tau = 1, 2, 3, 4$, and 5 were solved by both transition matrix and control parametrization approaches. The number of control parameters was chosen to be 10 in all five cases. The percentage difference of performance index values of the control parametrization approach relative to the transition matrix approach and the relative percent execution time are summarized in Table 3. It shows that the computational efficiency of the control parametrization approach improves slightly with the increase of τ. However, the error of the proposed approach increases significantly as the final time τ grows.

To study the accuracy of the control parametrization approach, simulations for the case of $\tau = 5$ were conducted using $J = 10$, 20, 30, 40 and 50. The results are summarized in Table 4. They confirm that improvement in accuracy of the proposed approach can be achieved by increasing the number of control parameters. Table 4 also shows that this improvement is achieved at the expense of increased computational cost. In particular, the proposed approach becomes less efficient than the transition matrix approach when J is equal to or larger than 30.

τ	% Error†	% Time‡
1	0·01	37·8
2	0·03	36·0
3	0·15	35·0
4	1·18	33·3
5	4·52	32·9

† Percentage difference of performance index of control parametrization approach relative to performance index value of transition-matrix approach.

‡ Percentage of execution time of control parametrization approach relative to execution time of transition-matrix approach.

Table 3. Summary of simulation results of Example 3 for problems with different τ.

Number of control parameters J	% Error†	% Time‡
10	4·52	32·9
20	1·15	65·4
30	0·51	115·0
40	0·29	171·8
50	0·18	250·9

† Percentage difference of performance index of control parametrization approach relative to performance index value of transition-matrix approach.

‡ Percentage of execution time of control parametrization approach relative to execution time of transition-matrix approach.

Table 4. Summary of simulation results of Example 3 for problems with different number of control parameters.

Results from Tables 3 and 4 suggest that the proposed approach may not be suitable for problems with large final time.

While problems with long final times may require a large number of control parameters in order to achieve high accuracy, it should be noted that the accuracy of the proposed approach may not necessarily be critical. The control parametrization approach is promoted as a simulation tool for predicting the system responses efficiently and thus useful for assisting designers in selecting appropriate weighting matrices. With the completion of the design stage, approaches such as a Riccati equation solver can be adopted for closed-loop implementation. It is at this implementation stage that the accuracy of the LQ control law is determined. Hence, the solution obtained from the control parametrization approach can be considered satisfactory provided that it predicts the general trend of the true optimal solution.

Table 3 indicates that for the final times investigated, the case of $\tau = 5$ and $J = 10$ has the largest error (4·52%) in performance index value. For this case, the control variable history of the parametrization solution is compared to the optimal control variable history in Fig. 2. In addition, the time histories of the state variable x_{10} obtained by the transition matrix approach and the proposed approach are compared in Fig. 3. These figures demonstrate that the control parametrization approach successfully predicts the pattern of the optimal control u and optimal state variable x_{10}. Although not presented, simulation results for the other nine state variables show a similar match of the proposed approach to the optimal solution.

4. Discussion

The computational requirements of control parametrization can be divided into three parts. The first part is related to the conversion from the continuous time model to a discrete time model (i.e. (4)). This process requires the evaluation of the transition matrix. The second part of the computational cost involves the construction of the performance index PI* (defined by (15)) and the corresponding necessary condition of optimality (defined by (18)). In particular, the terms defined in (16) have to be evaluated. The last part of the computation cost is incurred in solving the necessary condition of optimality.

The development of an optimal LQ control law typically involves adjustments of the state and control weighting matrices. In the standard approach, changing the

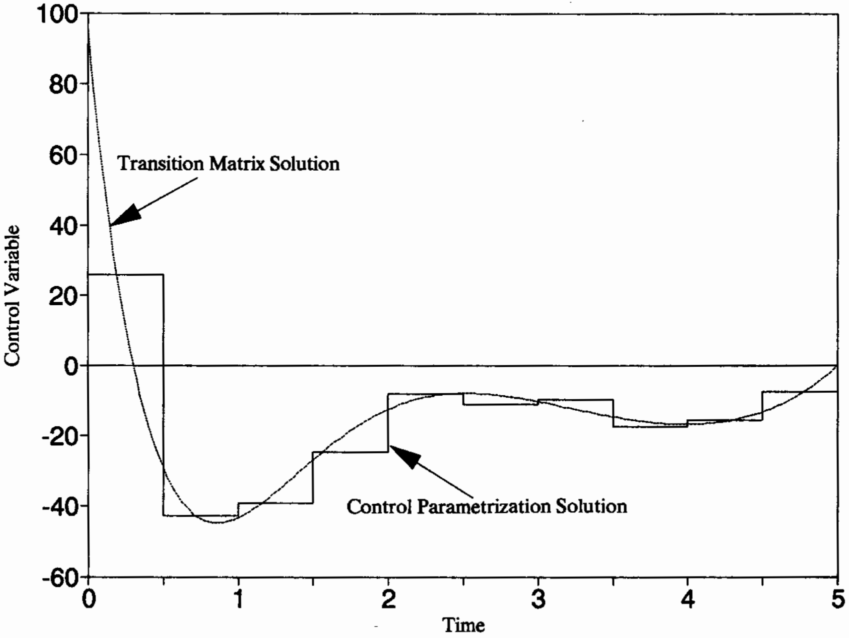

Figure 2. Control variable history for Example 3.

elements of the state and/or control weighting matrices requires a complete repetition of the solution process. These multiple solutions in the LQ design process can be partially avoided by control parametrization. In particular, the first and second part of the computations (described in the previous paragraph) can be eliminated. Consequently, the computational requirement for constructing the per-

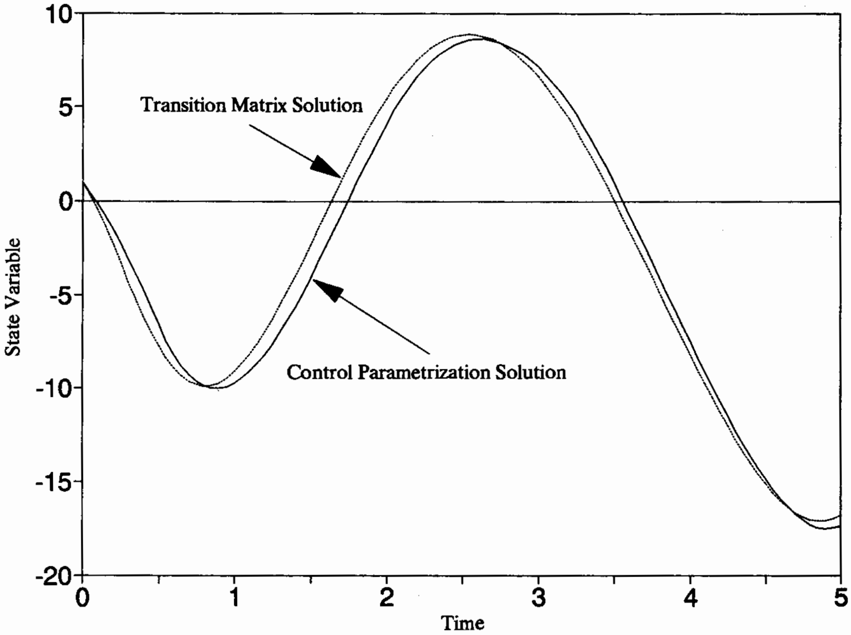

Figure 3. State variable x_{10} history for Example 3.

formance index PI* and the necessary condition of optimality can be reduced. In addition, when changes are made only to the diagonal terms of the control weighting matrix, the matrix inverse updating technique can be used to minimize the computational cost.

The simulation results show that control parametrization is a very competitive methodology in terms of computational efficiency when compared to the standard approach while maintaining satisfactory accuracy. The results demonstrate especially impressive efficiency in handling problems with small control/state dimension ratios. The practical payoff promises to be substantial since the number of control variables in most real, large-scale systems is only a small fraction of the system order. For such problems, the proposed approach would be a very useful tool.

One potential problem of the proposed approach is related to the selection of the number of control parameters. Clearly, the accuracy of the control approach increases with the number of control parameters. Unfortunately, since the optimal values of the control parameters are solved from a system of linear algebraic equations, the computational cost increases (cubically) with the dimension of the control parameter vector. As a result, the accuracy and the computational cost become competing (trade-off) factors in applying control parametrization. This conflict is particularly important for problems with large final time. However, as suggested in Example 3, the accuracy of the control parametrization is not critical provided that the proposed approach successfully predicts the general trend of the optimal solution.

A possible future direction of this work is to develop a more sophisticated parametrization scheme. For example, cubic splines could possibly provide a better approximation than the unit step function based parametrization method adopted in this paper. If so, the number of control parameters could be reduced without sacrificing accuracy. Another possible future direction is to generalize this approach for closed-loop systems. Currently, this possibility is under investigation and once developed will be compared with solution approaches for closed-loop LQ problems based on the Riccati equation.

5. Conclusion

This paper develops a control parametrization approach for solving LQ problems. Using control parametrization, the LQ problem is cast as a quadratic programming problem from which the necessary condition of optimality is derived as a system of linear algebraic equations.

As demonstrated by simulation results, the control parametrization approach achieves high accuracy and is particularly efficient in handling high order problems with small numbers of control variables. By minimizing the repetitive part of the computations, the approach can very efficiently solve a series of LQ problems with differences only in weighting matrices. Hence, the proposed approach promises to be an effective design tool for optimal LQ control law development.

References

Bashein, G., and Enns, M., 1972, *Int. J. Control*, **15**, 177.

Brewer, J. W., 1978, *I.E.E.E. Trans. Circuits Systems*, **25**, 772.

Dorato, P., and Levis, A. H., 1971, *I.E.E.E. Trans. autom. Control*, **16**, 613.

Gill, P. E., Golub, G. H., Murray, W., and Saunders, A., 1974, *Math. Comput.*, **28**, 505.

GILL, P. E., and MURRAY, W., 1977, *The State of the Art in Numerical Analysis* (New York: Academic Press), Chapter I.2.

GOH, C. J., and TEO, K. L., 1988, *Automatica*, **24**, 3.

GOLUB, G. H., and VAN LOAN, C. F., 1989, *Matrix Computations* (Baltimore, Md.: Johns Hopkins University Press), Chapter 4.

HICKS, G. A., and RAY, W. H., 1971, *Can. J. Chem. Engng*, **49**, 522.

NAGURKA, M. L., and YEN, V., 1990, *A.S.M.E. J. dynam. Systems Meas. Control*, **112**, 17.

NEUMAN, C. P., and SEN, A., 1973, *Automatica*, **9**, 601.

SIRISENA, H. A., and CHOU, F. S., 1981, *Optim. Control Applic. Methods*, **2**, 289.

SPEYER, J. L., 1986, *Control dynam. Systems: Adv. Theory Applic.*, **23**, 241.

VLASSENBROECK, J., and VAN DOOREN, R., 1988, *I.E.E.E. Trans. autom. Control*, **33**, 333.