Proc Indian Natn Sci Acad, **69**, **A**, Nos. 3 & 4, May & July 2003, pp. 301-315

TIME OPTIMAL CONTROL, POLE SHIFTING AND SIMULTANEOUS STABILIZATION

REIN LUUS

Department of Chemical Engineering, University of Toronto, Toronto, ON M5S 3E5 (Canada)

(Received 17 June 2002; Accepted 04 September 2002)

Time optimal control problems, where the desired state is to be reached from some given initial state, are generally very difficult to solve. In many cases the optimal control policy exhibits switching from one bound on the control variable to another bound, giving a bang-bang control policy. The switching times must be determined accurately in order to reach the desired state. Such a control policy is quite unstable and difficult to implement. There is the further problem of what to do once the desired state is reached, especially if the desired state is unstable in absence of control. To reach the desired state in a stable manner, shifting the poles of the linearized system may be used. It is not always clear if a satisfactory control policy is obtained since pole shifting is determined from a linearized system and the bounds placed on the control variables are not taken into account when the gain matrix is determined. Therefore, once the gain matrix is obtained, through simulation the viability of the control policy is established. If the parameters in the system are not known accurately, or if the parameters change, then we have the problem of simultaneous stabilization, where the same gain matrix is used to control a number of systems.

This paper outlines recent developments in time optimal control, pole shifting, and simultaneous stabilization and illustrates these with several examples.

Key Words: Time Optimal Control; Iterative Dynamic Programming; Pole Shifting; Simultaneous Stabilization; LJ Optimization Procedure

1 Introduction

Time optimal control problems are common in engineering practice. In the startup of a chemical plant, usually we want to reach the desired state of operation as quickly as possible. When going from one state of operation to another, one may wish to make the transition in minimum time, especially if the intermediate product has smaller economic benefit. In loading and unloading containers from a ship, the transfer of containers is usually desired in minimum time, but with the assurance of stability such time optimal control problems are frequently very difficult to solve, even with our present-day computers.

Often it may be desirable to approach the desired state a little bit slower, but in a stable fashion. At other times the parameters of the system may change, so that the time optimal control policy established for one set of system parameters may not be a good control policy once the parameters have changed, even very slightly. Therefore, one may find some time suboptimal control policy[1] more attractive. Thus in applying time optimal control in practice, one should also investigate procedures based on Lyapunov's direct method[2,3], pole shifting[4,5] and also simultaneous stabilization[6, 7] to ensure that a sufficiently robust control policy is obtained.

In this article we outline the latest developments in time optimal control, pole shifting and simultaneous stabilization, and illustrate the approaches with several examples.

2 Time Optimal Control

Let us consider the system described by the differential equation

$$\frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x}, \mathbf{u}) \qquad \ldots(1)$$

where the initial state $\mathbf{x}(0)$ is given. The state vector $\mathbf{x}$ is an $(n \times 1)$ vector and $\mathbf{u}$ is an $(m \times 1)$ control vector bounded by

$$\alpha_j \leq u_j \leq \beta_j,\ j = 1, 2, \cdots, m. \qquad \ldots(2)$$

At the final time t_f the state variables must be at the desired values x_i^d, $i = 1, \cdots, n$, i.e.,

$$h_i = x_i(t_f) - x_i^d = 0,\ i = 1, 2, \cdots, n. \qquad \ldots(3)$$

The time optimal control problem is to find the control policy $\mathbf{u}(t)$ in the time interval $0 \leq t < t_f$ that minimizes the performance index

$$I = t_f \qquad \ldots(4)$$

Numerically, it is required that in a minimum value of the final time t_f, we satisfy the constraint

$$|x_i(t_f) - x_i^d| \leq \varepsilon_i, \; i = 1, 2, \cdots, n, \qquad \ldots(5)$$

where ε_i is some tolerance, such as the measurement error. To solve this problem with the use of Pontryagin's maximum principle is very difficult, even for a low dimensional problem[8]. If eq. (1) is linear, however, then the problem can be solved quite readily by linear programming[9,10]. Recently it has been shown that iterative dynamic programming (IDP) provides an attractive alternative for solving time optimal control problems[11, 12].

To solve this time optimal control problem by IDP, we follow the suggestion of Luus[13] to choose the augmented performance index to be minimized as

$$J = t_f + \theta \sum_{i=1}^{n} (x_i(t_f) - x_i^d - s_i)^2, \qquad \ldots(6)$$

and update the shifting terms s_i after every pass of IDP according to

$$s_i^{p+1} = s_i^p - (x_i(t_f) - x_i^d), \qquad \ldots(7)$$

where p is the pass number in IDP.

To measure the closeness of the final states to the desired states we use the 1-norm, i.e., the sum of absolute deviations from the desired state, namely

$$S = \sum_{i=1}^{n} |x_i(t_f) - x_i^d|. \qquad \ldots(8)$$

In order to solve the time optimal control problem by IDP, we transform the continuous control policy into a piecewise constant control problem by dividing the time interval $[0, t_f]$ into P stages, each of variable length,

$$v(k) = t_k - t_{k-1}, \; k = 1, 2, \cdots, P. \qquad \ldots(9)$$

Computationally, we impose the condition

$$v(k) > 0, \; k = 1, 2, \cdots, P, \qquad \ldots(10)$$

so that we do not deal with negative or zero stage lengths.

Based on the idea of Bojkov and Luus[14], we transform the time variable through the relationship

$$dt = v(k)P d\tau \qquad \ldots(11)$$

so that in normalized time τ all the time stages are of equal length

$$\tau_k = \frac{k}{P}, \; k = 1, 2, \cdots, P. \qquad \ldots(12)$$

The differential equation in the time interval $\tau_{k-1} \leq \tau < \tau_k$ becomes

$$\frac{d\mathbf{x}}{d\tau} = v(k)P\mathbf{f}(\mathbf{x}, \mathbf{u}). \qquad \ldots(13)$$

With this transformation, at $t = t_f$, the transformed time variable $\tau = 1$, and the time to reach the desired state is

$$t_f = \sum_{k=1}^{P} v(k). \qquad \ldots(14)$$

Minimization of the augmented performance index J in eq. (6) should give the minimum value for t_f and will ensure that the state variables at the final time are close to the desired values. Such an approach has been found useful for a number of time optimal control problems (see ref. [15], Chapter 12).

3 Pole Shifting

Let us linearize eq. (1) with respect to the desired state into the form

$$\frac{d\mathbf{x}}{dt} = \mathbf{Ax} + \mathbf{Bu} \qquad \ldots(15)$$

where $\mathbf{A}$ is a constant $(n \times n)$ state matrix and $\mathbf{B}$ is a constant $(n \times m)$ matrix. Let us assume that the measured output is given by

$$\mathbf{y} = \mathbf{Cx}, \qquad \ldots(16)$$

where $\mathbf{y}$ is a $(q \times 1)$ output vector with $q \leq n$, and $\mathbf{C}$ is a constant $(q \times n)$ matrix.

The problem is to find a constant ($m \times q$) dimensional matrix $\mathbf{K}$, where

$$\mathbf{u} = \mathbf{K}\mathbf{y}, \qquad \ldots(17)$$

such that the real part of the least stable eigenvalue of the matrix

$$\mathbf{E} = \mathbf{A} + \mathbf{BKC} \qquad \ldots(18)$$

is minimized, subject to the constraints on each element of $\mathbf{K}$ in the form

$$\alpha_{ij} \leq k_{ij} \leq \beta_{ij}. \qquad \ldots(19)$$

Therefore, the performance index to be minimized is

$$I = \max_j Re[\lambda_j], \; j = 1, 2, \cdots, n, \qquad \ldots(20)$$

where λ_j is the j^{th} eigenvalue of the matrix $\mathbf{E}$. The eigenvalues of the matrix $\mathbf{E}$ are called poles, and during optimization, the pole that has the largest real part is shifted to the left. For that reason the method is called pole shifting.

Luus and Mutharasan[4] used two examples to show that such shifting of eigenvalues not only stabilizes the system, but yields control action which drives the system to the origin very fast. This led to an effective two-step procedure by Oh and Luus[5] to yield time-suboptimal control of engineering processes. In the first step, direct search was used to obtain the elements in a feedback gain matrix[16]. Once the vicinity of the desired state was reached, a switch was made to the gain matrix obtained by shifting the eigenvalues. This two-step procedure was shown to yield rapid, but stable, approach to the desired state[5]. Due to the recent developments in iterative dynamic programming (IDP)[15], for the first step, to get to the vicinity of the desired state in minimum time, we may use IDP rather than direct search.

Algorithm for Pole Shifting by LJ Optimization Procedure

For pole shifting we have a wide range of optimization procedures that may be used successfully. Recently, genetic algorithms have become popular for optimization of complex systems. However, an algorithm based on the one given by Luus and Mutharasan[4] using LJ optimization procedure[17,18] is straightforward and is easy to use. It is reliable and involves a small number of steps:

1. Select initial values for each k_{ij}; denote these by $k_{ij}^{*(0)}$.
2. Pick an initial region size for each k_{ij}; for example, we may pick a region corresponding to k_{ij} to be $r_{ij}^{(0)} = \beta_{ij} - \alpha_{ij}$. Set the iteration index l to 1.
3. Choose R different sets of values for the elements of the matrix $\mathbf{K}$ by choosing the elements

$$k_{ij} = k_{ij}^{*(l-1)} + d_{ij} r_{ij}^{(l-1)} \qquad \ldots(21)$$

where d_{ij} is a random number between -1 and 1.

4. For each feasible $\mathbf{K}$ evaluate the n eigenvalues of $\mathbf{E}$.
5. Find the $\mathbf{K}$ for which the maximum real part of the eigenvalues is minimum. Denote this by $\mathbf{K}^*$.
6. Increment the iteration index l by 1 to $l + 1$ and reduce the region size

$$r_{ij}^{(l)} = \gamma r_{ij}^{(l-1)}. \qquad \ldots(22)$$

7. Go to step 3; continue for a number of iterations and interpret the results.

This algorithm may be used in a multi-pass fashion, where we repeat the procedure a number of times, each time restoring the region size to a fraction of its size at the beginning of the previous pass. Such a multi-pass approach usually requires less computational effort to obtain the optimum value for the performance index than taking a very large number of random points and using a single pass[19].

4 Simultaneous Stabilization

To ensure robust control, we wish to have the same feedback matrix that could be used to control the system in stable manner even when the parameters in the system change. Therefore, the feedback gain matrix should be suitable for the entire range of variation of the parameters. For each set of extreme values of parameters, we have a separate system, giving us a set of systems corresponding to the "corner elements" in the parameter space. To obtain a single feedback gain matrix which stabilizes all the systems presents a simultaneous stabilization problem.

Simultaneous stabilization has been considered by numerous researchers including Howitt and Luus[6, 7], Zhang and Blondel[20], Wang and Fairman[21], Paskota *et al.*[22], Gundes and Kabuli[23], Liu *et al.*[24], Fernandez-Anaya *et al.*[25], Cao *et al.*[26], Das[27], Fonte *et al.*[28], Jia and Ackermann[29], Miller and Rossi[30], Saif *et al.*[31], and Huang and Huang[32]. Here we consider simultaneous

stabilization by the use of pole shifting and formulate the problem as follows:
We are given r sets of equations

$$\frac{d\mathbf{x}_k}{dt} = \mathbf{f}_k(\mathbf{x}_k, \mathbf{u}_k),\ k = 1, 2, \cdots, r, \qquad \ldots(23)$$

where $\mathbf{x}_k$ is an $(n \times 1)$ state vector for system k and $\mathbf{u}_k$ is an $(m \times 1)$ control vector, and the measured output is given by

$$\mathbf{y}_k = \mathbf{C}_k\mathbf{x}_k, \qquad \ldots(24)$$

where $\mathbf{y}_k$ is a $(q\times 1)$ output vector with $q \leq n$. The problem is to find a constant $(m\times q)$ dimensional matrix $\mathbf{K}$, where

$$\mathbf{u}_k = \mathbf{K}\mathbf{y}_k, \qquad \ldots(25)$$

subject to the constraints on each element of $\mathbf{K}$ in the form

$$\alpha_{ij} \leq k_{ij} \leq \beta_{ij}, \qquad \ldots(26)$$

such that all the r systems approach the origin in fast but stable manner.

If the systems described by eq. (23) are nonlinear, we linearize the systems into the form

$$\frac{d\mathbf{x}_k}{dt} = \mathbf{A}_k\mathbf{x}_k + \mathbf{B}_k\mathbf{u}_k,\ k = 1, 2, \cdots, r, \qquad \ldots(27)$$

and try to find a constant $(m\times q)$ dimensional matrix $\mathbf{K}$, such that the real part of the least stable eigenvalue of the matrix

$$\mathbf{E}_k = \mathbf{A}_k + \mathbf{B}_k\mathbf{K}\mathbf{C}_k \qquad \ldots(28)$$

is minimized. The real part of every eigenvalue of each $\mathbf{E}_k$ must be negative for a stabilizing control to exist.

To stabilize all of the systems, it is convenient to choose a combined performance index

$$I = -(I_1 + I_2 + \cdots + I_r) \qquad \ldots(29)$$

where for the k^{th} system

$$I_k = \frac{1}{\max_j Re[\lambda j]},\ j = 1, 2, \cdots, n, \qquad \ldots(30)$$

with the restriction

$$I_k < 0,\ k = 1, 2, \cdots, r. \qquad \ldots(31)$$

We, therefore, concentrate on the dominant eigenvalue, defined as the eigenvalue with the maximum real part, of each system. If the real part of the dominant eigenvalue of each system cannot be made negative, then all the systems cannot be stabilized by the single feedback gain matrix. The goal is to choose $\mathbf{K}$ to minimize I, and to interpret the results.

If there are constraints on the control, and the use of eq. (25) causes a control constraint to be violated, then we simply use the constraint value. Such clipping technique is widely used in optimal control.

5 Examples

All computations were performed on a PentiumII/400 personal computer in double precision using WATCOM FORTRAN 77 version 9.5 compiler. The built-in compiler random number generator URAND was used for generation of the random numbers. In order to illustrate and to test the computational viability, we consider several examples. In all the examples we used the DVERK[33] subroutine, listed in the Appendix of reference[15], to integrate the differential equations.

6 Brachistochrone Problem

The well-known brachistochrone (Gk. "minimum time") problem has been considered by many mathematicians and motivated the development of calculus of variations. Given two points A and B where B is lower than A, it is required to find the path that a bead, sliding along a frictionless wire under the influence of gravity only, should take in going from the point A to B in minimum time. This problem can be solved analytically by using the calculus of variations and has been presented in some undergraduate text books[34]. This time optimal control problem is interesting because the control policy is not bang-bang in nature, and has provided challenges for numerical procedures[35-37]. Here we shall solve this time optimal control problem with the use of iterative dynamic programming[15].

Let us pick point A as the origin, and consider the bead falling vertically in positive y direction and horizontally in the positive x direction. Let us choose the lower point B as (1.000, 0.500), where the distances are in m and the acceleration due to gravity is $g = 9.80\text{m/s}^2$.

Since friction is neglected, the conservation of energy provides the relationship for the speed of the bead in terms of the vertical distance fallen, i.e.,

$$\upsilon = \sqrt{2gy}. \qquad \ldots(32)$$

Therefore, the motion of the bead is described in terms of the two differential equations:

$$\frac{dx_1}{dt} = \sqrt{2gx_2}u \quad \text{...(33)}$$

$$\frac{dx_2}{dt} = \sqrt{2gx_2}\sqrt{1 - u^2} \quad \text{...(34)}$$

with the initial state

$$\mathbf{x}(0) = [0 \ \ 0]^T. \quad \text{...(35)}$$

Here x_1 is the horizontal distance from the point A, and x_2 is the vertical distance measured downward from the point A.

The control u is the cosine of the angle from the horizontal. Thus u should be smooth and continuous, and is constrained by $|u| \leq 1$. Thus a natural choice is to choose u to be piecewise linear and continuous. Here, however, we consider also the case where u is discontinuous; i.e., we will also consider piecewise constant control over short time stages as has been done by Teo *et al.*[37].

It is noted that the initial condition given in eq.(35) presents a singularity, where the right hand sides of both of the differential equations are zero. Therefore, we consider the problem in two parts.

First we consider the time t_1 for the bead to fall vertically downward 1 mm, i.e., to reach $x_2 = 0.001$; and then use the initial state:

$$\mathbf{x}(0) = [0 \ \ 0.001]^T \quad \text{...(36)}$$

for the second part of the trajectory. It is clear that the use of eq.(36) as an initial condition causes no numerical diffculties. It is easily seen that the time to fall 1 mm from rest is

$$t_1 = \sqrt{2 \times 0.001/9.8} = 0.01429\text{s}. \quad \text{...(37)}$$

The second part of the problem is to find the control u with the constraint $|u| \leq 1$ in the time interval $0 \leq t < t_2$ such that the desired state

$$\mathbf{x}^d = [1.000 \ \ 0.500]^T \quad \text{...(38)}$$

is reached from the initial state given by eq.(36) in minimum time t_2. The minimum time for the original problem is then simply $t_1 + t_2$.

To solve this problem with iterative dynamic programming (IDP)[15], we choose the augmented performance index to be minimized

$$J = t_2 + \theta[(x_1(t_2) - 1.0 - s_1)^2 + (x_2(t_2) - 0.5 - s_2)^2] \quad \text{...(39)}$$

where θ is a positive penalty function factor, and the shifting terms s_1 and s_2 are chosen initially zero and then are updated after every pass by

$$s_1^p = s_1^{p-1} - (x_1(t_2) - 1.0) \quad \text{...(40)}$$

$$s_2^p = s_2^{p-1} - (x_2(t_2) - 0.5). \quad \text{...(41)}$$

where p is the pass number. In the augmented performance index, a wide range of θ may be used to get accurate results. Here we choose the penalty function factor θ to be 1. The objective then is to choose u in the time interval $0 \leq t < t_2$ to minimize J. We consider the brachistochrone problem first without state constraints and then consider also the situation with state constraints.

(a) No State Constraints inside the Time Interval

We chose $P = 20$ time stages of variable length and used piecewise linear control. For the IDP algorithm we used $R = 25$ randomly chosen points at each time stage, and 50 passes of 10 iterations

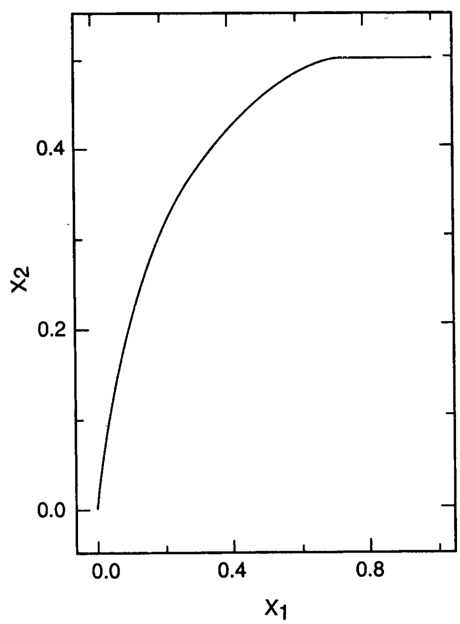

Fig. 1 Minimum time trajectory without constraints on the trajectory

each, and a single grid point at each time stage. The initial values for control u were chosen as 0.707, corresponding to 45° angle and each time stage was initially chosen to be of length 0.035 s. The initial region sizes for control and stage lengths were chosen as 0.2 and 0.01, respectively. After each iteration the region sizes were reduced by the reduction factor γ = 0.70, and after every pass, the region size was restored to η = 0.95 times the value at the beginning of the previous pass. In a computation time of 34.7 s on a PentiumII/400 personal computer for the 50 passes, a minimum time t_2 = 0.55605 was obtained. The sum of absolute deviations from the desired state was $S = 1.77 \times 10^{-5}$, giving a final time of t_f = 0.57034 s. The resulting trajectory is given in Fig. 1 and the control policy is given in Fig. 2.

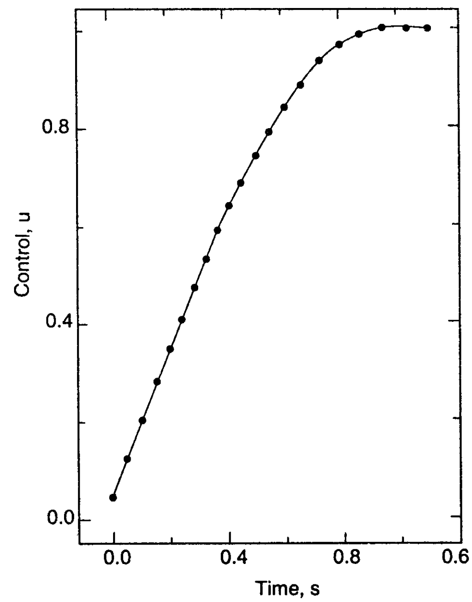

Fig. 2 Piecewise linear continuous control policy

The use of $P = 40$ stages of piecewise constant control gives t_2 = 0.55609 s with the deviation from the desired state $S = 2.78 \times 10^{-5}$. Therefore, piecewise linear control is substantially better than the larger number of piecewise constant control.

(b) Time Optimal Control with State Constraints

Here we consider the state constraint used by Bryson and Ho[36] and by Teo *et al.* [37], where we have the state constraint

$$0.5x_1(t) + 0.1 \geq x_2(t) \qquad \text{...(42)}$$

throughout the trajectory. To handle this state constraint, as suggested by Mekarapiruk and Luus[38], we introduce an auxiliary variable x_3 which is chosen zero initially and

$$\frac{dx_3}{dt} = x_2 - 0.5x_1 - 0.1 \qquad \text{...(43)}$$

when the state constraint is violated, and

$$\frac{dx_3}{dt} = 0 \qquad \text{...(44)}$$

when there is no violation. Therefore, if there is a state constraint violation, x_3 has a positive value. To discourage violation, we therefore include the additional term in the augmented performance index, i.e.,

$$J = t_2 + \theta_1[(x_1(t_2) - 1.0 - s_1)^2 + (x_2(t_2) - 0.5 - s_2)^2] + \theta_2 x_3(t_2) \qquad \text{...(45)}$$

where θ_1 is chosen to be 1 as before and θ_2 is chosen sufficiently large. Here we chose θ_2 = 100.

The use of 20 stages of piecewise linear control yielded t_2 = 0.57242 s with the sum of absolute deviation from the desired state of $S = 4.43\times10^{-5}$, with the control policy given in Fig. 3. Since x_3 (t_2) was zero, there was no state constraint violation, as can be seen also in Fig. 4. The use of 40 stages of piecewise constant control yielded slightly better results: t_2 = 0.57221 s and sum of deviations from desired state $S = 4.204\times10^{-6}$. The use of 100 stages of piecewise constant control provided a slight

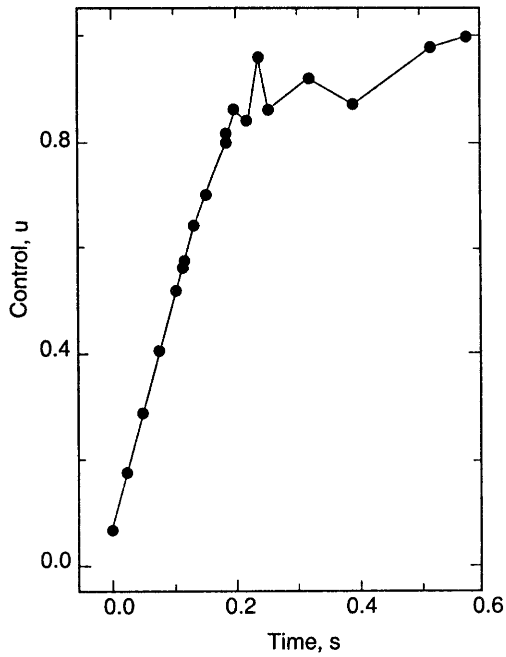

Fig. 3 Piecewise linear control policy for the brachistochrone problem with state constraint giving t_2 = 0.57242 s

improvement in the time to the vicinity of the desired state to $t_2 = 0.57186$s and sum of deviations of $S = 3.48 \times 10^{-5}$. The control policy resulting from the use of 100 time stages is given in Fig. 5 and the resulting trajectory is shown in Fig. 6.

Let us now consider the direct path to the desired state. This can be obtained by imposing the constraint

$$x_2 \leq 0.5x_1 \qquad \ldots(46)$$

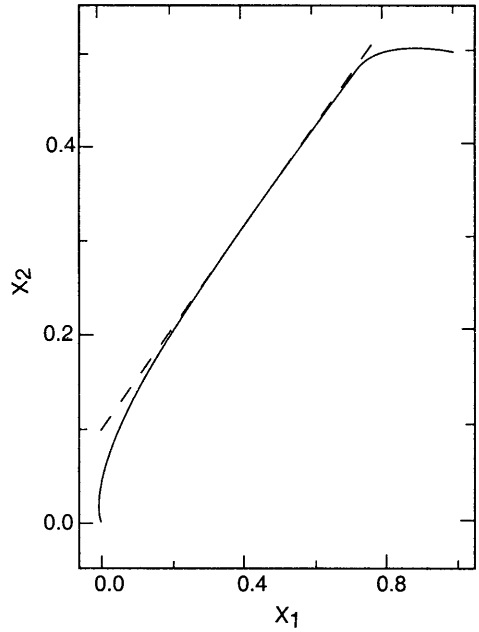

Fig. 4 Minimum time trajectory with constraint (dashed line) on the trajectory

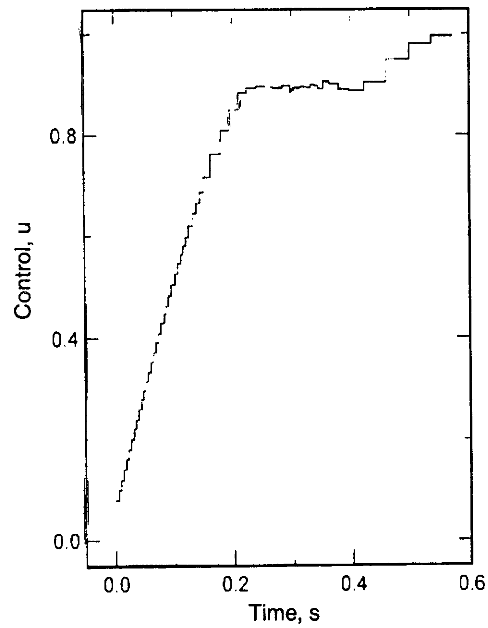

Fig. 5 Piecewise constant control policy for brachistochrone problem with constraint on the trajectory

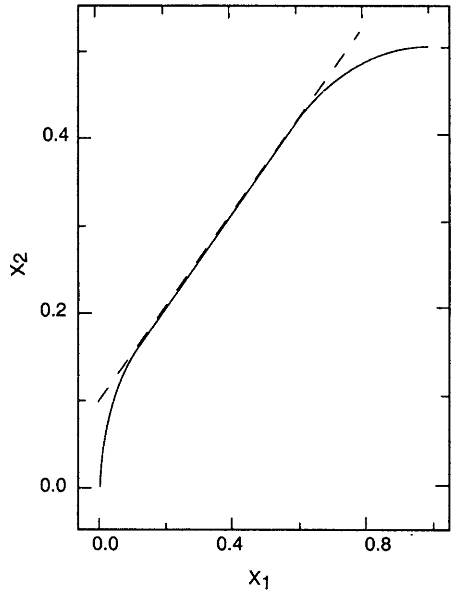

Fig. 6 Minimum time trajectory with constraint on the trajectory (dashed line) resulting from the use of the control policy in Fig. 5

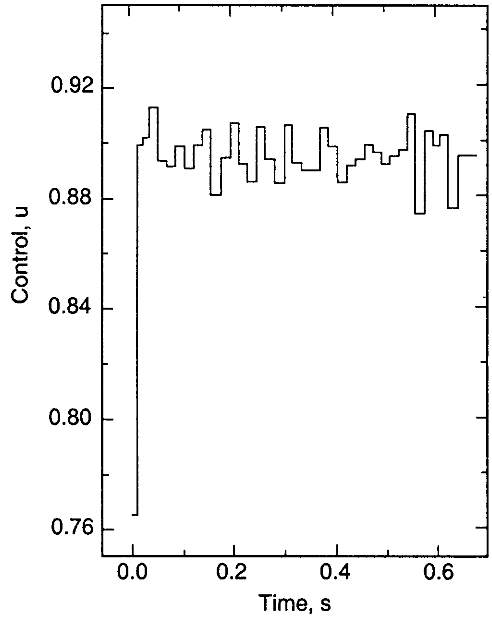

Fig. 7 Minimum time control policy for the brachistochrone problem with constraint $x_2 \leq 0.5x_1$

To avoid any difficulties near the origin, we choose as the initial state

$$\mathbf{x}(0) = [0.002 \quad 0.001]^T . \qquad \ldots(47)$$

The use of $P = 40$ time stages and piecewise constant control yielded $t_2 = 0.68154$ s with sum of deviations of $S = 3.01 \times 10^{-5}$. The resulting control policy is given in Fig. 7 and the trajectory is given in Fig. 8. It is seen that the time taken here is substantially longer than in the unconstrained case.

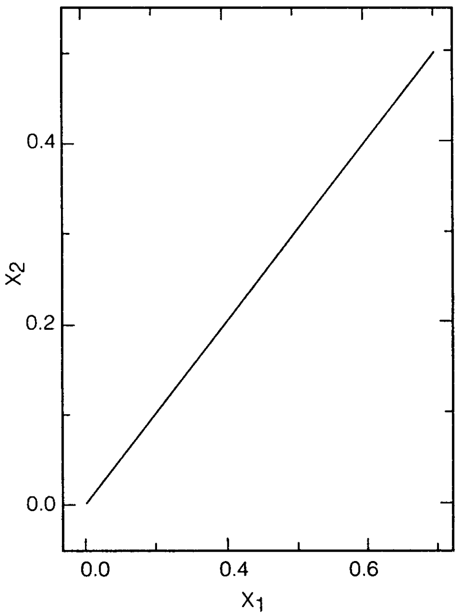

Fig. 8 Trajectory resulting from the control policy in Fig. 7

7 Six-Stage Extraction Train

Let us consider the nonlinear Six-stage extraction train where there is a feed to the third stage, as first used for optimal control studies by Koepcke and Lapidus[2], outlined by Lapidus and Luus[3], and used for time suboptimal control by Luus [39]. The equations describing the system are

$$\frac{dx_1}{dt} = -2.32x_1 + 1.32x_2 + (0.006665 - x_1)u_1 \quad \ldots(48)$$

$$\frac{dx_2}{dt} = x_1 - 2.32x_2 + 1.32x_3 + (0.005050 + x_1 - x_2)u_1 \quad \ldots(49)$$

$$\frac{dx_3}{dt} = x_2 - 2.82x_3 + 1.32x_4 + (0.003825 + x_2 - x_3)u_1 + (0.00540 - x_3)u_2 \quad \ldots(50)$$

$$\frac{dx_4}{dt} = 1.5x_3 - 2.82x_4 + 1.32x_5 + (0.004998 + x_3 - x_4)(u_1 + u_2) \quad \ldots(51)$$

$$\frac{dx_5}{dt} = 1.5x_4 - 2.82x_5 + 1.32x_6 + (0.005678 + x_4 - x_5)(u_1 + u_2) \quad \ldots(52)$$

$$\frac{dx_6}{dt} = 1.5x_5 - 2.82x_6 + (0.006453 + x_5 - x_6)(u_1 + u_2) \quad \ldots(53)$$

with the initial state

$$\mathbf{x}^T(0) = [-0.008784 - 0.009584 - 0.007975 - 0.008937 - 0.007774 - 0.004741]. \quad \ldots(54)$$

The state variable x_i represents the deviation from the desired steady-state concentration in stage i, and u_j represents the deviation from steady-state of the normalized flow rate, with the bounds

$$-1 \leq u_j \leq 2, \ j = 1, 2. \quad \ldots(55)$$

The problem is to bring the system from the initial state to the origin defined by $|x_i(t_f)| < 0.0001$, $i = 1, 2, \cdots, 6$ in minimum time.

By using IDP with $P = 10$ time stages of varying length, $\gamma = 0.95$, $\eta = 0.95$, $R = 15$ points per iteration, 200 passes, each consisting of 20 iterations, with $\theta = 2.5\times10^4$, and initially zero for control, we obtained in 51.9 s on a PentiumII/400 a minimum value of $t_f = 0.97155$, $S = 2.4175 \times 10^{-4}$ with the control policy given in Table I. At the final time the state was $\mathbf{x}^T = [3\ -4\ 3\ -4\ 6\ -5] \times 10^{-5}$, so the final state criterion is satisfied. As can be seen from the Table I, the optimal control policy consists of only four time stages, where at the first time stage of length 0.37704 $u_1 = 2$ and $u_2 = -1$. It was found that for this system the time of switching does not have to be very precise. Even approximate switching times take the system to the origin very fast. By using the control policy: $u_1 = 2$, $u_2 = -1$ until $t = 0.38$, then $u_1 = 2$, $u_2 = 2$ until $t = 0.70$, $u_1 = -1$, $u_2 = 2$ until $t = 0.79$, and then $u_1 = -1$, $u_2 = -1$ until $t = 0.97$ brings the state to $\mathbf{x}^T = [8\ -1\ 2\ -7\ 3\ -6] \times 10^{-5}$ with $S = 2.6 \times 10^{-4}$ at $t = 0.97$.

Now let us apply pole shifting to this problem. To linearize the equations, we simply drop the bilinear terms to give the state equations in the form

$$\frac{d\mathbf{x}}{dt} = \mathbf{Ax} + \mathbf{Bu} \quad \ldots(56)$$

where

$$\mathbf{A} = \begin{bmatrix} -2.32 & 1.32 & 0 & 0 & 0 & 0 \\ 1 & -2.32 & 1.32 & 0 & 0 & 0 \\ 0 & 1 & -2.82 & 1.32 & 0 & 0 \\ 0 & 0 & 1.5 & -2.82 & 1.32 & 0 \\ 0 & 0 & 0 & 1.5 & -2.82 & 1.32 \\ 0 & 0 & 0 & 0 & 1.5 & -2.82 \end{bmatrix} \quad \ldots(57)$$

and

$$\mathbf{B} = \begin{bmatrix} 0.006665 & 0 \\ 0.005050 & 0 \\ 0.003825 & 0.005540 \\ 0.004998 & 0.004998 \\ 0.005678 & 0.005678 \\ 0.006453 & 0.006453 \end{bmatrix} \quad \ldots(58)$$

Now we seek the control $\mathbf{u}$ in the form

$$\mathbf{u} = \mathbf{Kx} \quad \ldots(59)$$

where the feedback gain matrix **K**

$$\mathbf{K} = \begin{bmatrix} k_{11} & k_{12} & \cdots & k_{16} \\ k_{21} & k_{22} & \cdots & k_{26} \end{bmatrix} , \qquad \ldots(60)$$

is chosen to shift the dominant eigenvalues of

$$\mathbf{E} = \mathbf{A} + \mathbf{BK} \qquad \ldots(61)$$

as far to the left as possible. By using LJ optimization with the algorithm given earlier, we obtained

$$\mathbf{K} = \begin{bmatrix} -289.43 & -417.00 & -281.53 & 198.08 & 280.19 & 26.95 \\ 405.99 & -344.73 & -417.00 & -417.00 & -294.63 & -106.42 \end{bmatrix} , \ldots(62)$$

giving the eigenvalues:

$$\lambda = \begin{bmatrix} -4.17181 \pm 0.03248i \\ -4.17181 \pm 0.39777i \\ -4.17181 \pm 0.41661i \end{bmatrix} . \qquad \ldots(63)$$

It is interesting to note that all the eigenvalues have the same real part.

Table I

Time Optimal Control Policy for the Extraction Train Obtained by IDP

Stage No.	Control u_1	Control u_2	Stage length
1	2.000	-1.000	0.09795
2	2.000	-1.000	0.13776
3	2.000	-1.000	0.14133
4	2.000	2.000	0.04887
5	2.000	2.000	0.04317
6	2.000	2.000	0.08731
7	2.000	2.000	0.07295
8	2.000	2.000	0.06761
9	-1.000	2.000	0.09578
10	-1.000	-1.000	0.17884

Comparison of the approach to the origin using pole shifting to time optimal control is shown in Fig. 9. The approach with the feedback gain matrix in eq. (62) is actually faster initially than with time optimal control policy, but it takes until $t = 1.86$ min for all state variables to be less than 0.0001 in magnitude. In the absence of control the system is stable, but it takes considerably longer to reach the vicinity of the origin.

It is interesting to note that in using the feedback gain matrix in eq. (62), the initial section of the control policy is quite different from the time optimal control policy, as is shown in Fig. 10. Here both u_1 and u_2 start off at their maximum values, rather than having u_2 initially at its minimum value as is shown in Table I.

Typical state trajectories for time optimal control and the use of control from pole shifting are shown in Figs. 11 and 12 respectively.

To approach the origin fast but in a stable manner, one could use the time optimal control policy up to a given time and then switch to the feedback gain

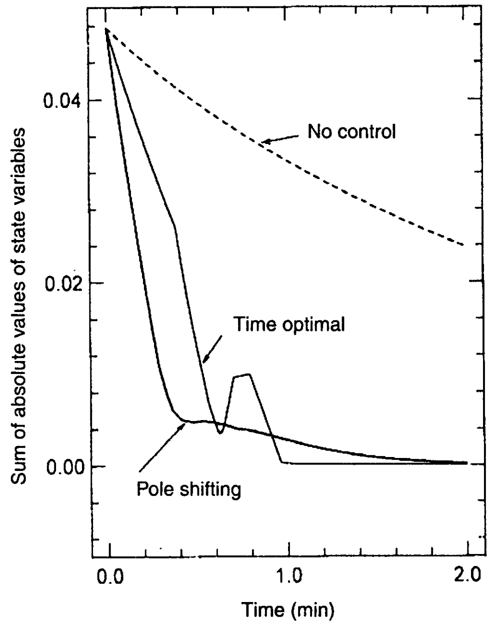

Fig. 9 Comparison of pole shifting to time optimal control for the extraction train

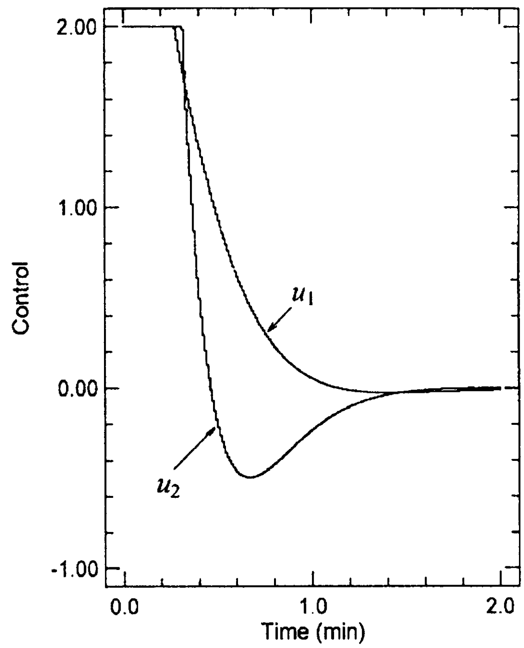

Fig.10 Control policy for the extraction train obtained with pole shifting

matrix. Such a two-pass method was suggested by Oh and Luus[5]. To illustrate its use here, we used the time optimal control policy up to $t = 0.62$ and then switched to the feedback gain matrix. The approach to the origin is shown in Figs. 13 and 14. The origin is reached in t = 1.62 min in a stable manner, so it is faster than using pole shifting alone.

This system is easy to control because its steady state is stable and the switching times are not very sensitive. To show a more challenging situation, we consider the 2-CSTR system next.

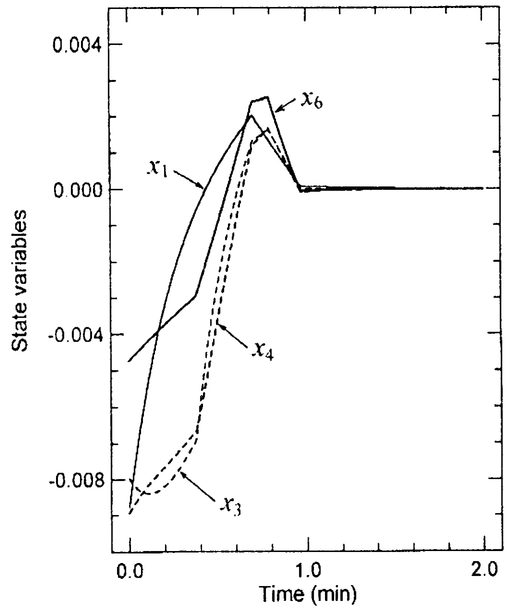

Fig. 11 Approach to the origin of typical state variables of the extraction train with the use of time optimal control

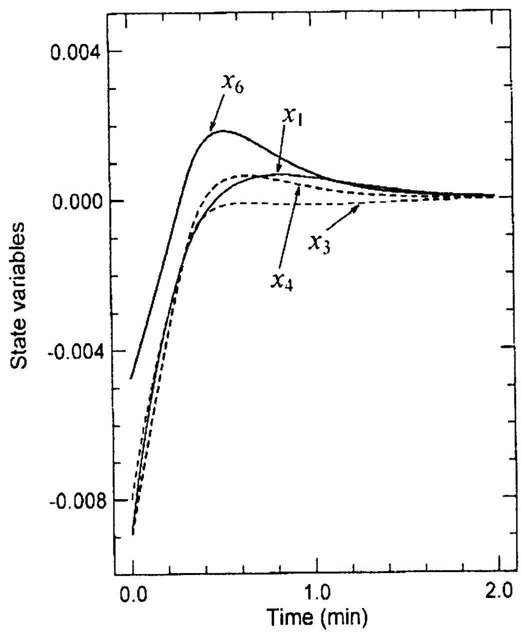

Fig. 12 Approach to the origin of typical state variables with use of pole shifting

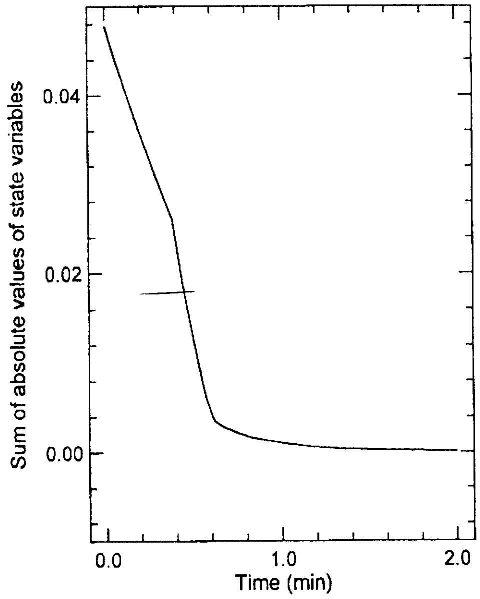

Fig. 13 Two-pass method, where time optimal control policy is used up to $t = 0.62$ and then the control policy is calculated from the feedback gain matrix obtained by pole shifting

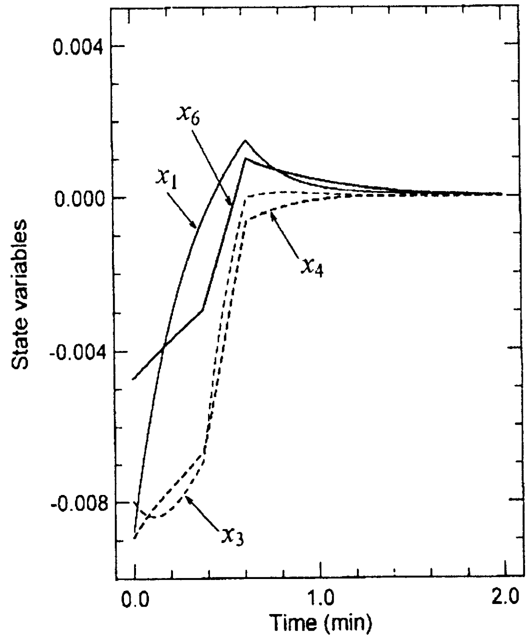

Fig. 14 Approach to the origin of typical state variables of the extraction train with the two-step procedure

8 Nonlinear Two-Stage CSTR System

Let us consider the two-stage CSTR (continuous stirred tank reactor) system used by Oh and Luus[5] to illustrate a two-step procedure for time suboptimal control. The system is described by

$$\frac{dx_1}{dt} = 0.5 - x_1 - R_1 \qquad \ldots(64)$$

$$\frac{dx_2}{dt} = -2(x_2 + 0.25) - u_1(x_2 + 0.25) + R_1 \quad \ldots(65)$$

$$\frac{dx_3}{dt} = x_1 - x_3 - R_2 + 0.25 \quad \ldots(66)$$

$$\frac{dx_4}{dt} = x_2 - 2x_4 - u_2(x_4 + 0.25) + R_2 - 0.25 \quad \ldots(67)$$

where x_1 and x_3 are normalized concentration variables in tanks 1 and 2 respectively, and x_2 and x_4 are normalized temperature variables in tanks 1 and 2 respectively. The reaction rate term in the first tank is given by

$$R_1 = (x_1 + 0.5) \exp\left(\frac{25x_2}{x_2 + 2}\right) \quad \ldots(68)$$

and in the second tank

$$R_2 = (x_3 + 0.25) \exp\left(\frac{25x_4}{x_4 + 2}\right) \quad \ldots(69)$$

The normalized controls are bounded by

$$-1 \leq u_j \leq 1, \; j = 1, 2 \quad \ldots(70)$$

and the initial state is

$$\mathbf{x}(0) = [-0.15 \; 0.05 \; - 0.10 \; 0.02]^T. \quad \ldots(71)$$

Here we consider the time optimal control, pole shifting and simultaneous stabilization related to this system.

The time optimal control policy is readily obtained by IDP with the use of $P = 10$ time stages, giving the bang-bang control policy in Table II. The time to the origin is $t_f = 0.84475$ with $S = 9.62 \times 10^{-8}$. At the final time all $x_i = 0.00000$.

Thus the time optimal control policy consists of only four time stages as given in Table III, giving $t_f = 0.8448$.

If, however, we try to implement the time optimal control policy, we run into difficulties because we must choose some sampling time. Let us first choose the sampling time to be 0.005. Initially we keep $u_1 = 1$, $u_2 = 1$ until $t = 0.275$, then switch to $u_1 = -1$, $u_2 = 1$ and keep the control at these values until $t = 0.335$ when the control is switched to $u_1 = -1$, $u_2 = -1$. Then at $t = 0.785$ we switch to $u_1 = 1$, $u_2 = -1$. At $t = 0.845$ we expect to have reached the origin and switch to $u_1 = 0$, $u_2 = 0$. With this control policy, we find that at $t = 0.845$, $\mathbf{x} = [-0.00190 \; 0.00283 \; - 0.00074 \;\; 0.00127]^T$, putting the state into the vicinity of the origin, but not very close to it. As is shown in Fig. 15, the system starts to diverge quite rapidly after the control is put to zero at $t = 0.850$.

Table II

Time Optimal Control Policy for the 2-stage CSTR System Obtained by IDP

Stage No.	Control u_1	Control u_2	Stage length
1	1.000	1.000	0.01697
2	1.000	1.000	0.00608
3	1.000	1.000	0.06567
4	1.000	1.000	0.05197
5	1.000	1.000	0.07073
6	1.000	1.000	0.06419
7	-1.000	1.000	0.06104
8	-1.000	-1.000	0.17307
9	-1.000	-1.000	0.27295
10	1.000	-1.000	0.06209

Table III

Time Optimal Control Policy for the 2-stage CSTR System

Stage No.	Control u_1	Control u_2	Stage length
1	1.000	1.000	0.27563
2	-1.000	1.000	0.06104
3	-1.000	-1.000	0.44602
4	1.000	-1.000	0.06209

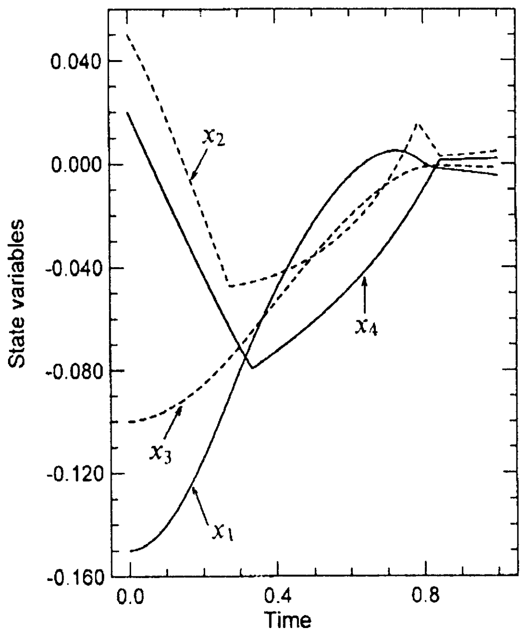

Fig. 15 Approach to the origin of the state variables of the 2-stage CSTR system when time optimal control policy is used with a sampling time of 0.005

By using a smaller sampling time of 0.001, and using the switches at $t = 0.276, 0.337, 0.783$, and 0.845, we can get closer to the desired state as is shown in Fig. 16. It is therefore obvious that it is very difficult to implement the exact switching times, and for this system the response is very sensitive to the time of switching. We now turn to pole shifting.

The linearized system is

$$\frac{d}{dt}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} -2 & -6.25 & 0 & 0 \\ 1 & 4.25 & 0 & 0 \\ 1 & 0 & -2 & -3.125 \\ 0 & 1 & 1 & 1.125 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ -0.25 & 0 \\ 0 & 0 \\ 0 & -0.25 \end{bmatrix}\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \quad ...(72)$$

We assume that only the temperature of each tank is measured. Then the control is expressed as

$$\mathbf{u} = \mathbf{K}\mathbf{y} \quad ...(73)$$

where

$$\mathbf{y} = \begin{bmatrix} x_2 \\ x_4 \end{bmatrix}, \quad ...(74)$$

and the feedback matrix is

$$\mathbf{K} = \begin{bmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{bmatrix}, \quad ...(75)$$

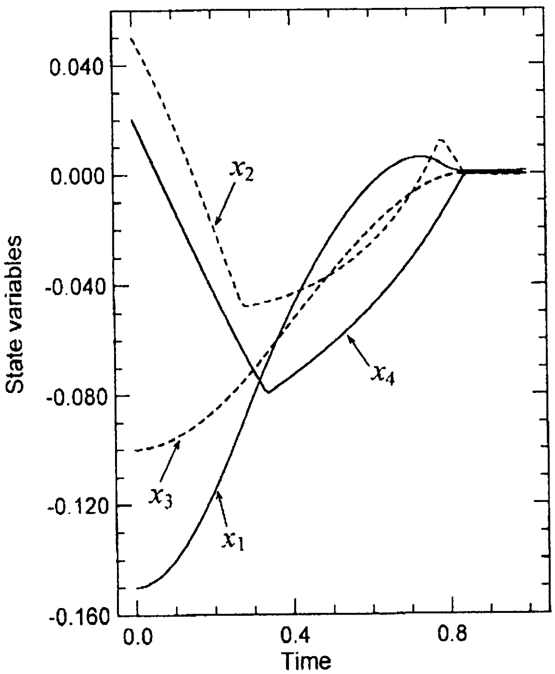

Fig. 16 Approach to the origin of the state variables of the 2-stage CSTR system when time optimal control policy is used with a sampling time of 0.001

Thus

$$\mathbf{E} = \begin{bmatrix} -2 & -6.25 & 0 & 0 \\ 1 & (4.25 - 0.25k_{11}) & 0 & -0.25k_{12} \\ 1 & 0 & -2 & -3.125 \\ 0 & (1 - 0.25k_{21}) & 1 & (1.125 - 0.25k_{22}) \end{bmatrix} \quad ...(76)$$

The performance index we wish to minimize is

$$I = \max_j Re[\lambda_j], \quad j = 1, 2, \cdots, 4, \quad ...(77)$$

where λ_j is the j^{th} eigenvalue of the matrix $\mathbf{E}$ given in eq. (76).

For optimization in shifting the eigenvalues of $\mathbf{E}$ as far left as possible, we put 62.5 as the upper limit for the size of the elements of $\mathbf{K}$ and used a single pass consisting of 201 iterations. By using an initial value of $\mathbf{0}$ for $\mathbf{K}$, and initial region size of 62.5, with region reduction factor $\gamma = 0.95$, we found that the resulting elements in $\mathbf{K}$ were dependent on the number of random values R chosen per iteration as is shown in Table IV. The computation time for the 201 iterations with $R = 10000$ was 404 s on a PentiumII/400 personal computer.

To simplify the control policy we made a run with $R = 1000$ by putting $k_{12} = 0$ and $k_{21} = 0$. We obtained $I = -3.76777$ with the resulting feedback gain matrix:

$$\mathbf{K} = \begin{bmatrix} 42.38 & 0 \\ 0 & 26.64 \end{bmatrix}, \quad ...(78)$$

giving the four eigenvalues $-3.76777 \pm 0.00028i$, $-4.17210 \pm 1.23774i$ for $\mathbf{E}$. Therefore, the use of a diagonal feedback gain matrix still gives very large negative real parts of the eigenvalues and fast response is expected. The use of this gain matrix gave fast and stable approach to the origin as is shown in Fig. 17, with the control policy in Fig. 18.

Table IV

Feedback Gain Elements Obtained by Pole Shifting for the 2-stage CSTR System

Random points	Performance index, I	Feedback gain elements k_{11}	k_{12}	k_{21}	k_{22}
100	-4.478	23.62	-13.36	41.65	53.53
1000	-4.586	17.42	-17.29	54.16	61.45
10000	-4.614	16.82	-17.63	55.95	62.50

Now let us consider the situation when the activation energy is not accurately known, or when it changes during the course of the operation of the 2-stage reactor

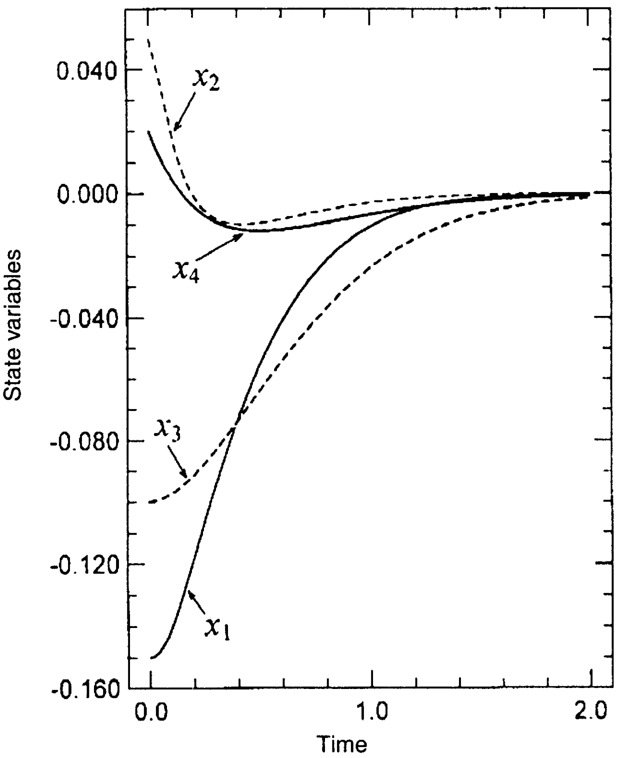

Fig. 17 Approach to the origin of the state variables of the 2-stage CSTR system when the feedback gain matrix in eq. (78) is used

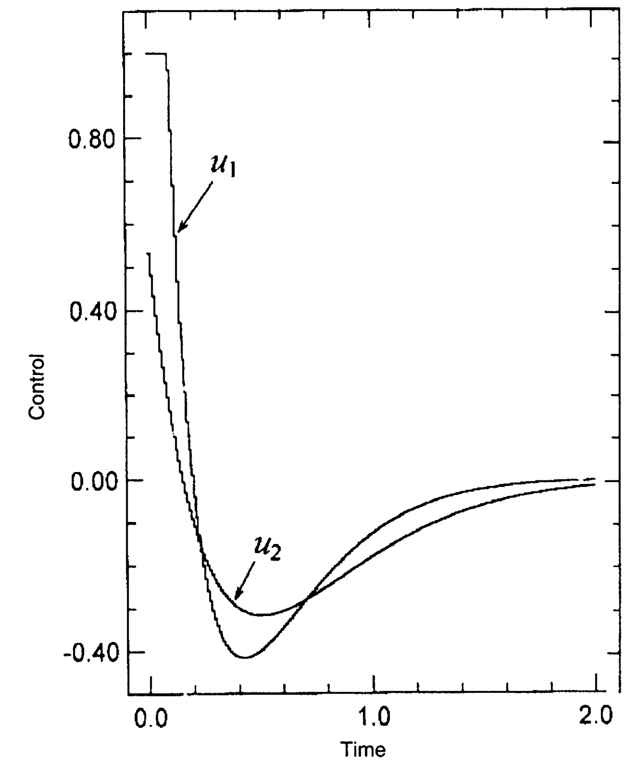

Fig. 18 Control policy for the 2-stage CSTR system obtained with the feedback gain matrix in eq. (78)

system. Thus, instead of considering the expected value $E = 25$, we consider two additional values for the extreme limits of variation. We therefore consider 3 systems simultaneously with $E_1 = 20$, $E_2 = 25$, and $E_3 = 30$, and seek a single feedback gain matrix that will ensure that all these systems could be stabilized. We also want the approach to the origin to be as fast as possible, so we consider the simultaneous stabilization problem from the optimization point of view as outlined earlier.

The nonlinear equations are linearized into the form

$$\frac{d\mathbf{x}_k}{dt} = \mathbf{A}_k\mathbf{x}_k + \mathbf{B}_k\mathbf{u}_k \qquad \ldots(79)$$

where the three systems based on the linearized equations can be written as

$$\frac{d}{dt}\begin{bmatrix} x_{1k} \\ x_{2k} \\ x_{3k} \\ x_{4k} \end{bmatrix} = \begin{bmatrix} -2 & -0.25E_k & 0 & 0 \\ 1 & (0.25E_k-2) & 0 & 0 \\ 1 & 0 & -2 & -0.125E_k \\ 0 & 1 & 1 & (0.125E_k-2) \end{bmatrix}\begin{bmatrix} x_{1k} \\ x_{2k} \\ x_{3k} \\ x_{4k} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ -0.25 & 0 \\ 0 & 0 \\ 0 & -0.25 \end{bmatrix}\begin{bmatrix} u_{1k} \\ u_{2k} \end{bmatrix} \qquad \ldots(80)$$

We assume that only the temperature of each tank is measured. Then the control is expressed as

$$\mathbf{u}k = \mathbf{K}\mathbf{y} \qquad \ldots(81)$$

where

$$\mathbf{y} = \begin{bmatrix} x_{2k} \\ x_{4k} \end{bmatrix} \qquad \ldots(82)$$

and the feedback matrix is the same for all the three systems, namely

$$\mathbf{K} = \begin{bmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{bmatrix} \qquad \ldots(83)$$

Thus

$$\mathbf{E}_k = \begin{bmatrix} -2 & -0.25E_k & 0 & 0 \\ 1 & (0.25E_k\text{-}2\text{-}\ 0.25k_{11}) & 0 & -0.25k_{12} \\ 1 & 0 & -2 & -0.125E_k \\ 0 & (1 - 0.25\ k_{21}) & 1 & (0.125E_k\text{-}2\text{-}0.25k_{22}) \end{bmatrix} \qquad \ldots(84)$$

The problem is to determine $\mathbf{K}$ to minimize the performance index

$$I = -(I_1 + I_2 + I_3) \qquad \ldots(85)$$

where for the k^{th} system

$$I_k = \frac{1}{\max_j Re[\lambda_j]} \quad j = 1, 2, \cdots, 4, \qquad \text{...(86)}$$

with the restriction

$$I_k < 0, \; k = 1, 2, 3. \qquad \text{...(87)}$$

What makes this problem particularly interesting is that in absence of control the three systems are unstable, having positive dominant eigenvalues of 1.618, 3.000 and 4.312 respectively, and therefore, in absence of control, the systems will drift further from the starting value as is shown in Fig. 19.

By using $R = 100$ and 100 passes, each consisting of 21 iterations, we obtained $I = 0.86918$ with

$$\mathbf{K} = \begin{bmatrix} 43.7557 & -0.6038 \\ 0.5583 & 27.7205 \end{bmatrix} \qquad \text{...(88)}$$

As is shown in Fig. 20, the approach to the origin for all these three systems is quite systematic, and this feedback gain matrix not only stabilizes all of the systems, but brings the systems to the origin quite rapidly.

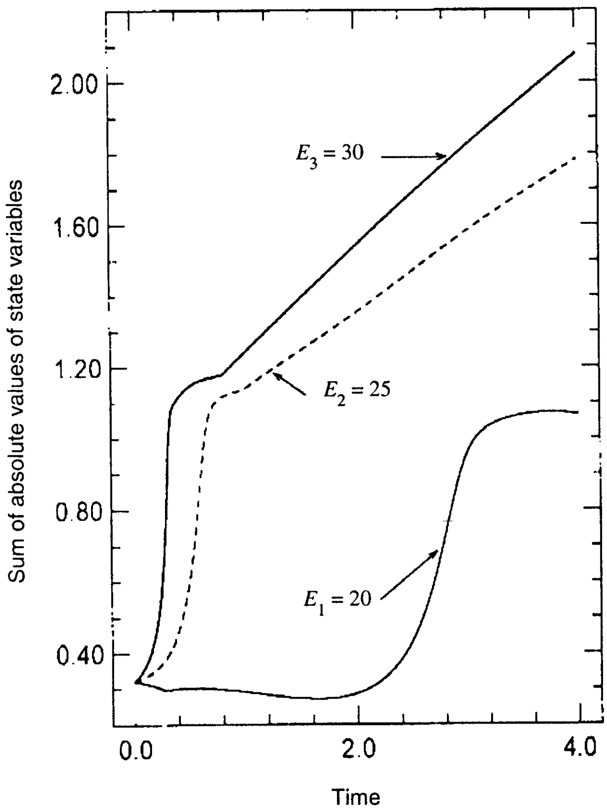

Fig. 19 Sum of absolute values of the four state variables for the three systems with $\mathbf{K} = \mathbf{0}$

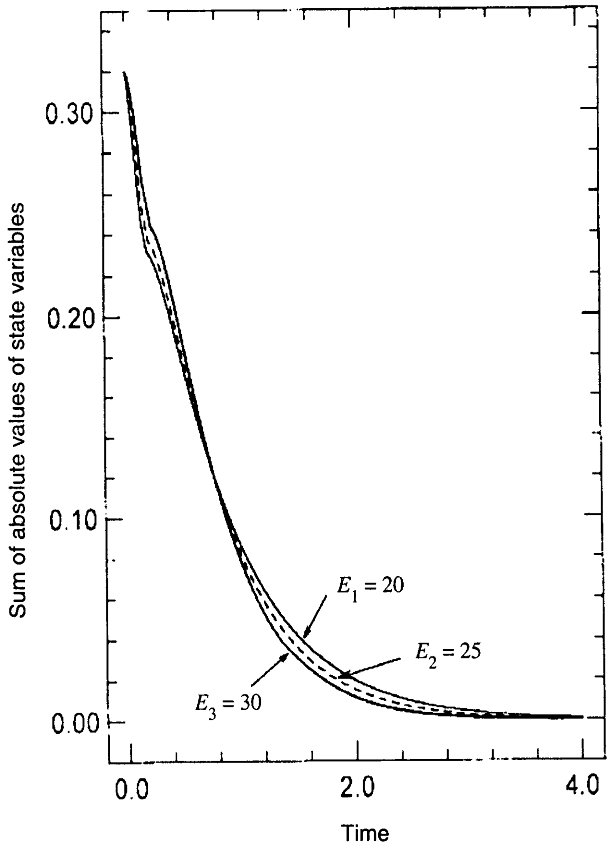

Fig. 20 Sum of absolute values of the four state variables for the three systems with $\mathbf{K}$ given in eq. (88).

Conclusions

As illustrated with the brachistochrone problem, time optimal control is not always bang-bang in nature. In many problems, however, the time optimal control policy is bang-bang which may be difficult to implement. Therefore pole shifting provides a useful way of reaching the desired state reasonably fast, but in a stable manner.

Iterative dynamic programming provides a convenient and reliable means of solving the brachistochrone problem. Numerical difficulties associated with the singularity were avoided by solving the problem in two parts, showing that there is always room for ingenuity and one must not rely on numerical procedures alone. The use of a quadratic penalty function with shifting terms is a good means of handling the final state constraint, and allows the final state to be reached arbitrarily close. The state constraint inside the time interval was easily handled by introducing an additional state variable as suggested by Mekarapiruk and Luus[38].

To provide robust control in the presence of parameter variations, simultaneous stabilization approach was used where the dominant eigenvalues of each system were shifted to the left as far as possible. Numerical results show the practicality of such an approach.

Acknowledgement

Financial support from the Natural Sciences and Engineering Research Council of Canada is gratefully appreciated.

References

1 R Luus Suboptimal Control *Encyclopedia of Optimization* (Eds. C A Floudas and P M Pardalos) Kluwer Academic Publishers Dortrecht The Netherlands **5** (2001) 384

2 R Koepcke and L Lapidus *Chem Eng Sci* **16** (1961) 252

3 L Lapidus and R Luus *Optimal Control of Engineering Processes* Blaisdell (1967) 354

4 R Luus and R Mutharasan *Int J Control* **20** (1974) 395

5 S H Oh and R Luus *Int J Control* **22** (1975) 749

6 G D Howitt and R Luus *Int J Control* **54** (1991) 1015

7 G D Howitt and R Luus *Int J Control* **58** (1993) 79

8 S M Moon and H F VanLandingham, On the Variational Approach in a Time Optimal Control Problem *Proc IASTED International Conf CONTROL'97* Cancun Mexico (May 1997) 290

9 G Bashein *IEEE Trans Aotomat Contr* **16** (1971) 479

10 O Rosen, Imanudin and R Luus *Int J Control* **45** (1987) 1371

11 B Bojkov and R Luus *Ind Eng Chem Res* **33** (1994) 1486

12 B Bojkov and R Luus *Can J Chem Eng* **73** (1995) 380

13 R Luus Direct Approach to time Optimal Control by Iterative Dynamic Programming *Proc IASTED International Conf Intelligent Systems and Control* June 1-4, Halfax Nova Scotia Canada (1998) 121

14 B Bojkov and R Luus *Chem Eng Sci* **51** (1996) 905

15 R Luus *Iterative Dynamic Programming* London UK Chapman & Hall/CRC (2000)

16 R Luus *Chem Eng Sci* **29** (1974) 1013

17 R Luus and T H I Jaakola *AIChE J* **19** (1973) 760

18 R Luus Luus-Jaakola Optimization Procedure *Recent Res Devel Chemical Engg* Transworld Research Network **4** (2000) 45

19 R Luus, F Hartig and F J Keil *Hungarian J Ind Chem* **23** (1995) 55

20 C S Zhang and V Blondel *Int J Control* **57** (1993) 293

21 S M Wang and F W Fairman *Int J Control* **59** (1994) 1095

22 M Paskota, V Sreeram, K L Teo and W Y Yan *Int J Control* **60** (1994) 483

23 A N Gundes and M G Kabuli *IEEE Trans Automat Contr* **40** (1995) 1926

24 W Q Liu, K L Teo and W Y Yan *Int J Control* **64** (1996) 595

25 G Fernandez-Anaya, S Munoz-Gutierrez R A Sanchez-Guzman and W W Mayol-Cuevas *Int J Control* **68** (1997) 1417

26 Y Y Cao, Y X Sun and J Lam *IEEE Trans Automat Contr* **44** (1993) 1277

27 S K Das *IEEE Trans Automat Contr* **46** (2001) 125

28 C Fonte, M Zasadzinski, C Bernier-Kazantsev and M Darouach *IEEE Trans Automat Contr* **46** (2001) 1101

29 Y M Jia and H Ackermann *Automatica* **37** (2001) 1425

30 D E Miller and M Rossi *IEEE Trans Automat Contr* **46** (2001) 1543

31 A W Saif, D W Gu, D Kavranoglu and I Postelethwaithe *IEEE Trans Automat Contr* **47** (2002) 363

32 X Huang and B Huang *Dynam Cont Dis and Imp Systems Ser B* **9** (2002) 71

33 T E Hull, W D Enright and K R Jackson *User Guide to DVERK- a Subroutine for Solving Nonstiff ODE's*. Report 100 Department of Computer Science University of Toronto Canada (1976)

34 L Page *Introduction to Theoretical Physics* D Van Nostrand Co Inc. New York NY Third Edition (1952) 204

35 R E Bellman and S E Dreyfus *Applied Dynamic Programming* Princeton University Press Princeton, NJ (1962) 327

36 A E Bryson and Y C Ho *Applied Optimal Control* MA Waltham Ginn and Company (1969) 119

37 K L Teo, C J Goh and K H Wong *A Unified Computational Approach to Optimal Control Problems* John Wiley & Sons Inc New York NY (1991) 155

38 W Mekarapiruk and R Luus *Ind Eng Chem Res* **36** (1997) 1686

39 R Luus *Ind Eng Chem Proc Des Develop* **13** (1974) 405