Part V. Quasilinearization and State-Constrained Systems

The quasilinearization algorithm is examined as a means of determining the optimal control of a nonlinear tubular reactor system with state constraints. It is shown that the use of penalty functions in an unconstrained algorithm can effectively solve the constrained problem. The method is quadratically convergent to the optimal control and seems more successful than methods which require gradients in the computation.

The application of the optimal control theory to dynamic systems which are characteristic of the chemical process industry is often hampered by the existence of constraints on the allowable system behavior, in addition to the multidimensional, nonlinear nature of such systems. Fortunately, many optimal control strategies will satisfy the state constraints on the system without explicitly bringing these restrictions into the problem. However, the converse occurs often enough to make the study of optimal systems in the presence of state constraints a real and pressing problem.

In the present paper modifications of the variational approach to unconstrained problems that provide necessary conditions for state-constrained problems are summarized. It is shown, for example, that the maximum principle yields rather cumbersome adjoint equations when the state space is bounded. Only one numerical procedure which explicitly satisfies these necessary conditions is currently available, this being an extension of the gradient method for unconstrained systems. In an effort to avoid the complexity of such direct methods, a penalty function technique is introduced that essentially incorporates the constraint into the performance index by means of a suitably chosen function. The function reduces to zero if the constraint is satisfied and penalizes the performance index if the constraint is violated. The constrained problem is thus converted into an unconstrained one. By increasing the weight of this penalty term relative to the performance index, it is possible to satisfy the constraint more and more exactly.

A quasilinearization algorithm which incorporates this penalty function method is also introduced. This technique generates an increasing sequence of weighting factors for the penalty term. Each weighting factor corresponds to an unconstrained problem, which is solved by quasilinearization. In short, by use of penalty functions, a sequence of unconstrained problems is solved by repeated application of the quasilinearization algorithm for unconstrained systems, and this sequence of solutions converges to the solution of the constrained problem in the limit.

To illustrate the solution of state-constrained problems by quasilinearization, the famous Amundson-Bilous tubular reactor problem (*4*) is altered by the addition of a quadratic state constraint. It is seen that the quasilinearization algorithm worked moderately well, although the violation of the constraint could not be reduced beyond a certain level. Nevertheless, this level was acceptable since the control policy obtained differed by less than 1% from the optimal policy. The IBM-7094 computation time required for this problem with two state variables varied between 6 and 25 min., depending on the maximum acceptable constraint violation. Based on this, larger systems may demand excessive computation time. It is generally concluded that quasilinearization with penalty functions provides a workable algorithm for the optimization of dynamic systems in the presence of state constraints, although excessive computation time may prohibit the treatment of very large systems.

B. F. Rothenberger is with Sun Oil Company, Marcus Hook, Pennsylvania.

NECESSARY CONDITIONS FOR STATE-CONSTRAINED SYSTEMS

In this paper we treat the same basic problem as in Part IV of this series (*18*). The system under discussion is given by Equations (1) to (3) of that paper, and in addition we require that the state vector be restricted by the scalar inequality relationship

$$g[\mathbf{x}(t)] \leq 0 \qquad (1)$$

for all t, $0 \leq t \leq t_f$. Before proceeding to a discussion of the solution of the current problem, we note several characteristics of the formulation.

1. While Equation (1) is an inequality constraint, equality constraints are easily handled in this formulation, since they merely reduce the dimension of the state vector.
2. Equation (1) can also be generalized to a vector of constraints, $\mathbf{g}[\mathbf{x}(t)] \leq \mathbf{0}$. Thus a scalar g is employed here without loss of generality.
3. The constraint function g depends only on the state of the system, since control does not appear explicitly in Equation (1). There are two reasons for considering only state constraints without explicit dependence on $\mathbf{u}(t)$. First, state-variable restrictions in practice are usually independent of control. Second, when $\mathbf{u}(t)$ appears explicitly in g, the necessary conditions for an optimum are considerably more tractable than if g is independent of $\mathbf{u}(t)$. Thus the present analysis treats the more difficult of the two cases and the one most frequently encountered in practice, state constraints explicitly independent of control.

We begin the attack on the state-constrained problem by developing the necessary conditions for the existence of an optimal control policy $\mathbf{u}^*(t)$, $0 \leq t \leq t_f$. The derivation of the necessary conditions for minimizing the performance index is basically an extension of the approach for unbounded systems (*17, 19*). For the unconstrained case we define a Hamiltonian function and an adjoint vector with final values prescribed [Equations (4), (5) and (6) of reference *18*]. In addition, $\mathbf{u}(t)$ is chosen at each time to maximize H. This corresponds to minimizing the performance index I.

The necessary conditions for systems with state constraints have been investigated by Dreyfus (*3, 10*), Berkovitz (*2, 3*), Bryson (*6*), Guinn (*11*), Chang (*7*), and Gamkrelidze (*17*). Berkovitz and Dreyfus (*3*) assert that the adjoint variables satisfy Equations (4) and (5) of reference *18* for any portion of $\mathbf{x}(t)$ which lies within the

constraint boundary. If $g(\mathbf{x}) = 0$, then the adjoint equations are, for a scalar control ($r = 1$)

$$\frac{dz_i}{dt} = -\frac{\partial H}{\partial x_i} + \nu \frac{\partial}{\partial x_i}(\nabla g \cdot \mathbf{F}), \quad i = 1, \ldots, n, \quad (2)$$

where

$$\nabla g \cdot \mathbf{F} = \sum_{i=1}^{n} \frac{\partial g(\mathbf{x})}{\partial x_i} F_i(\mathbf{x}, u) \quad (3)$$

and $\nu = \nu(t)$ is a continuous scalar function of t. In addition, maximizing H one gets

$$\sum_{i=1}^{n} z_i \frac{\partial F_i}{\partial u} + \nu \frac{\partial}{\partial u}(\nabla g \cdot \mathbf{F}) = 0 \quad (4)$$

which determines $\nu(t)$. At the point $t = t_s$ where $\mathbf{x}(t)$ passes from the region $g(\mathbf{x}) < 0$ to the boundary $g(\mathbf{x}) = 0$, either the jump condition

$$z_i^+(t_s) = z_i^-(t_s) + \mu \frac{\partial g}{\partial x_i}, \quad i = 1, \ldots, n, \quad (5)$$

or

$$z_i^-(t_s) + \mu \frac{\partial g}{\partial x_i} = 0, \quad i = 1, \ldots, n, \quad (6)$$

holds. $\mathbf{z}^+$ is the adjoint vector just after the junction with the boundary at $t = t_s$, $\mathbf{z}^-$ is the adjoint vector just before $t = t_s$, and μ is some real number. Note that the jump condition produces a multipoint, boundary-value problem.

The solution of the necessary conditions presented above yields the optimal policy $\mathbf{u}^*(t)$. We note, however, that numerically computing such a solution for a specific system is not an easy task. It is to this problem that we now turn our attention.

COMPUTATIONAL CONSIDERATIONS

We have already noted in a companion paper (*18*) that even the two-point boundary-value problem formed by unconstrained systems poses considerable numerical difficulties. Based on this experience it is reasonable to expect that any algorithm that attempts to calculate the state-constrained optimum by directly solving the necessary conditions will face many obstacles. The only workers to implement the necessary conditions in any form are Denham and Bryson (*8*). They modify the popular gradient method (*5*) for unconstrained systems. The application of the necessary conditions is somewhat easier in this case because the gradient method utilizes a linearized system as part of its implementation. Hence the adjoint equations given by Equation (3) are also linear, and this greatly simplifies the procedure. Denham and Bryson applied this direct method to a three-variable flight control problem with some success. They did, however, report difficulty in obtaining a reasonable initial control policy. One disturbing result was a discontinuity in the scalar control variable at the junction point t_s. This discontinuity probably does not exist in the true optimal policy. Denham and Bryson report convergence within ten iterations or less for this three-variable problem. Convergence was assumed when the performance index I was changing less than 0.2%. No computation times are given. Further comments on this direct algorithm will be made later.

In order to avoid the use of the necessary conditions for optimality of a control policy in the presence of state constraints, it is convenient to utilize an indirect method which incorporates the constraint into the performance index through a penalty function. In contrast to the direct method presented above, the penalty function algorithms generate a sequence of unconstrained problems whose solutions approach the solution to the constrained problem in the limit. The basic idea is to add to the performance index I a term which penalizes the control policy if it produces a violation of the state constraint. By successively increasing the weight of this term relative to I, the constraint is met more and more exactly and the sequence of solutions so generated hopefully approaches the solution to the constrained problem. Simplicity over the direct method is gained because this is a sequence of unconstrained solutions, each of which can be treated by the numerous methods for the optimization of unconstrained systems.

Okamura (*16*) and Russell (*20*) provide theoretical support for the penalty function techniques by deriving existence and convergence theorems. Okamura considers an augmented cost function $\tilde{J}_i$ given by

$$\tilde{J}_i = J(u) + \int_0^{t_f} \sigma_i \phi(g) dt, \quad i = 1, 2, \ldots, \quad (7)$$

where $J(u)$ is the unaugmented integral objective function

$$J(u) = \int_0^{t_f} B(\mathbf{x}, u) dt \quad (8)$$

$\phi(g)$ is the penalty function, defined such that

$$\phi(g) \begin{cases} > 0, & g(\mathbf{x}, u) > 0 \\ = 0, & g(\mathbf{x}, u) = 0 \end{cases} \quad (9)$$

and $\{\sigma_i\}$ is a sequence of penalty weighting coefficients such that $\sigma_i < \sigma_j$ if $i < j$. Okamura shows that as $i \to \infty$, $\tilde{J}_i$ approaches a limiting value which is indeed the optimal value of the unaugmented objective function. Although Okamura only considers constraints which depend explicitly on u, his theorems appear to apply just as well to state constraints. Russell (*20*) derives analogous results for systems which are linear in control.

Kelley (*13*) uses this penalty function device along with the method of gradients. He introduces a new state variable x_{n+1} into the system defined by

$$\dot{x}_{n+1} = [g(\mathbf{x})]^2 \cdot h[g(\mathbf{x})] \quad (10)$$

with the initial condition prescribed by

$$x_{n+1}(0) = 0 \quad (11)$$

In Equation (10) h is the Heaviside unit step function; that is

$$h = \begin{cases} 1, & g(\mathbf{x}) > 0 \\ 0, & g(\mathbf{x}) \leq 0 \end{cases} \quad (12)$$

Then it follows that the constraint violation is

$$x_{n+1}(t_f) = \int_0^{t_f} g^2 h(g) dt \quad (13)$$

The addition of this term to the objective function with an appropriate penalty weighting coefficient produces Equation (7). Hence the penalty function is

$$\phi(g) = g^2 h(g) \quad (14)$$

Kelley defines the tolerance as

$$E = \int_0^{t_f} \epsilon^2 h(g) dt \quad (15)$$

where ϵ is a specified error in the sense of a root-mean-square average over instantaneous values. The ratio of the violation $x_{n+1}(t_f)$ to the tolerance E provides a way of picking $\{\sigma_i\}$ during the gradient descent process. In a numerical example, Kelley increased σ_i in proportion to $|x_{n+1}(t_f)/E|$ to determine σ_{i+1}. For the first few iterations convergence was improved if σ_i was kept relatively small. It would appear that a considerable amount of guesswork is needed to determine $\{\sigma_i\}$.

Kelley notes that, from his experience with penalty functions and the gradient method, attempts to produce close tolerances frequently lead to oscillation. This is not unexpected since it is well known that the method of gradients for unconstrained systems is slow to converge in the neighborhood of the optimum (*5, 14*).

Denham and Bryson (*8*) claim that their direct method is superior to the gradient method with penalty functions

for three reasons: an additional state variable is not needed; each approximate solution using the direct method satisfies the constraint at least over a portion of the trajectory; and for periods during which the trajectory is on the constraint boundary no adjustments are made by the direct method, thus decreasing computation time.

They also applied the gradient method with penalty functions to the same three-variable flight control problem used to illustrate the direct method. The results indicated that more iterations were required when penalty functions were employed. However, neither method gave truly optimal control.

Denn and Aris (9) made use of the penalty function concept with the method of Green's functions to satisfy state variable constraints at t_f. While this is a somewhat different problem, the results are informative. They found that as σ_i increased, convergence became more and more difficult owing to increased sensitivity of the Green's vector to changes in $\mathbf{x}(t_f)$. This corroborates the observations of Kelley.

In summary, the main advantage of the penalty function approach appears to be simplicity; instead of solving the original difficult problem, a series of easier problems are solved. Penalty function algorithms also have disadvantages, however. The above discussion has exposed three potential limitations. First, the speed of convergence may be prohibitively slow as the optimum is neared. Second, the best sequence of penalty weighting coefficients is not readily apparent. Finally, the ability of penalty function algorithms to achieve close constraint violation tolerances is uncertain.

QUASILINEARIZATION

Quasilinearization with penalty functions is a numerical algorithm which overcomes the drawbacks of the penalty function methods detailed above while maintaining simplicity. It is well known that quasilinearization is a powerful iterative method for solving unconstrained, nonlinear, boundary-value problems. Its application to the optimal control of unconstrained chemical processes has already been demonstrated (*18*).

McGill (*15*) proposes the use of penalty functions and quasilinearization to solve the state-constrained control problem as follows. First, define a new state variable $x_{n+1}(t)$ by Equations (10) and (11), where h is given by Equation (12) as before. Applying the maximum principle for unconstrained systems, the adjoint variable associated with x_{n+1} is given by

$$\dot{z}_{n+1} = 0 \tag{16}$$

Equation (16) implies that $z_{n+1}(t) = z_c$, a constant. From the maximum principle $z_{n+1} < 0$, since H is being maximized. Denoting the unconstrained system Hamiltonian as H_o, the Hamiltonian for the state-constrained system, but with the addition of a penalty function, is

$$H = H_o + z_{n+1}\,[g(\mathbf{x})]^2 \cdot h[g(\mathbf{x})] \tag{17}$$

It is easily seen by comparison with Equation (7) that z_{n+1} is equivalent to the penalty weighting coefficient σ_i.

The penalty function boundary-value problem can now be stated. The $n + 1$ state equations are given by Equations (1) of reference *18* and (10), with initial conditions prescribed by Equations (3) of reference *18* and (11). The n equations adjoint to the state equations are given by Equation (5) of reference *18*, where H is the Hamiltonian defined by Equation (4) of *18*. The n final conditions on the adjoint equations are prescribed by Equation (6) of reference *18*. The $(n + 1)^{\text{th}}$ adjoint equation is Equation (16), where the boundary condition on z_{n+1} is the constant z_c. This, then, is the boundary value problem: $2n + 2$ first-order differential equations with $n + 1$ initial values and $n + 1$ final values. Note that nothing has been said as yet about the value of z_c, only its sign. The choice of z_c will be taken up next.

Based on the theorems of Okamura and Russell, which have already been mentioned, it is plausible to expect that there exists a sequence $\{z_c^{(m)}\}$ of z_c's to which there is associated a sequence $\{x_{n+1}^{(m)}(t_f)\}$ of constraint violations such that as $m \to \infty$, $x_{n+1}^{(m)} \to 0$. Having said this a method for generating a sequence of z_c's can be presented. McGill (*15*) proposes the use of a scalar form of the Newton-Raphson algorithm. This can be written as

$$|z_c^{(m+1)}| = |z_c^{(m)}| - \left\{\frac{|z_c^{(m)}| - |z_c^{(m-1)}|}{x_{n+1}^{(m)}(t_f) - x_{n+1}^{(m-1)}(t_f)}\right\} x_{n+1}^{(m)}(t_f) \tag{18}$$

Equation (18) predicts $z_c^{(m+1)}$ so that $x_{n+1}^{(m+1)}(t_f)$ will be reduced to zero, based on a linear extrapolation of the previous two members of the sequence. Since three indexes are present in Equation (18), some other way of getting $z_c^{(1)}$ must be found. McGill suggests

$$z_c^{(1)} = z_c^{(0)} + K x_{n+1}^{(0)}(t_f) \tag{19}$$

where K is some suitable constant, and the value of $z_c^{(0)}$ is usually zero. Equations (18) and (19) generate the sequence $\{z_c^{(m)}\}$.

At each value of m the nonlinear boundary-value problem developed above must be solved. McGill (*15*) proposes quasilinearization for this task. We have discussed the quasilinear algorithm in detail elsewhere (*18*). Briefly, the method makes use of the fact that linear boundary-value problems can be solved by the superposition of a particular solution and a series of weighted homogeneous solutions. Consequently we linearize the state and adjoint differential equations, Equations (1) and (5) of reference *18*, about the m^{th} trajectory and iteratively solve this linear boundary-value problem to yield a convergent series of approximation to the $\mathbf{x}^{(m+1)}$ and $\mathbf{z}^{(m+1)}$ trajectories. Convergence is assumed to have occurred if the maximum change in any element of the state or adjoint vector is no greater than some prescribed tolerance ϵ. This trajectory then becomes the $(m + 1)^{\text{th}}$ approximation to the constrained optimum and $x_{n+1}^{(m+1)}(t_f)$, the constraint violation, is determined by integrating Equation (10) with the newly determined values of $x^{(m+1)}(t)$ used. If $|x_{n+1}^{(m+1)}(t_f)|$ is less than some tolerance ϵ_c, computation ceases. If not, a new value of z_c, namely $z_c^{(m+2)}$, is chosen in accordance with Equation (18), with m replaced by $m + 1$, and iteration proceeds.

Before the development of the quasilinearization penalty function algorithm for state-constrained systems is completed, several points pertaining to the quasilinear method should be listed. First, for $m = 0$ an initial guess of $\mathbf{x}(t)$ and $\mathbf{z}(t)$ is required to start the iteration. It has been shown (*18*) that the region of feasible starting points, from which convergence to within a tolerance ϵ can be obtained, is bounded. However, this region is large enough so that sophisticated procedures are not necessary to provide a feasible starting point. Second, quasilinearization is hampered by the well-known instability of the adjoint equations to forward integration (*1, 17*). However, this limitation is not too severe, at least for unconstrained systems, and ways of partially circumventing the problem are available (*18*). By contrast, quasilinearization provides quadratic convergence in some cases, making it superior to other algorithms for unconstrained systems (*12, 14*).

It now remains to demonstrate the effectiveness of the algorithm for problems with state restrictions by applying

it to a numerical example. Specifically, the following questions will be answered.

1. How many iterations are required to satisfy a given ϵ? A given ϵ_c?
2. Does Equation (18) generate a sequence of penalty weighting coefficients $\{z_c^{(m)}\}$, which leads to rapid convergence of the algorithm?
3. How small a value of ϵ_c can be satisfied without excessive iteration?

A NUMERICAL EXAMPLE

Consider the reaction $A \rightarrow B \rightarrow C$ taking place in a tubular reactor. If the reactions are first order, material balances for A and B for plug flow are

$$\frac{dx_1}{d\tau} = -k_1 x_1 \tag{20}$$

$$\frac{dx_2}{d\tau} = k_1 x_1 - k_2 x_2 \tag{21}$$

where x_1 and x_2 are the mole fractions of A and B, respectively, at the point in the bed corresponding to a residence time τ. For a given feed

$$\mathbf{x}(0) = \mathbf{x}^o \tag{22}$$

it is desired to choose the temperature along the bed, such that the outlet mole fraction of B, $x_2(\tau_f)$, is maximized, where k_1 and k_2 are given by

$$k_i(\tau) = G_i \exp\left[-E_i/RT(\tau)\right], \quad i = 1, 2 \tag{23}$$

This is the so-called *Amundson-Bilous problem* (4). In addition the state of the system is constrained by

$$g(x_1, x_2) = \eta_1 x_1^2 + \eta_2 x_1 + \eta_3 x_1 x_2 + \eta_4 x_2 + \eta_5 x_2^2 + \eta_6 \leq 0 \tag{24}$$

When one includes this constraint by means of a penalty function and introduces a new state variable $x_3(\tau)$ defined by Equations (10) and (11), the adjoint equations are

$$\frac{dz_1}{d\tau} = k_1(z_1 - z_2) - z_3 h(2\eta_1 x_1 + \eta_2 + \eta_3 x_2) \tag{25}$$

$$\frac{dz_2}{d\tau} = k_2 z_2 - z_3 h(2\eta_5 x_2 + \eta_4 + \eta_3 x_1) \tag{26}$$

where

$$z_1(\tau_f) = 0$$
$$z_2(\tau_f) = -1 \tag{27}$$

and $h = h[g(\mathbf{x})]$, defined by Equation (12). Minimizing the Hamiltonian

$$H = -k_1 x_1 z_1 + (k_1 x_1 - k_2 x_2) z_2 + z_3 h(g)\, g(x_1, x_2) \tag{28}$$

with respect to T, since we are maximizing $I = x_2(\tau_f)$, we get the optimal temperature profile

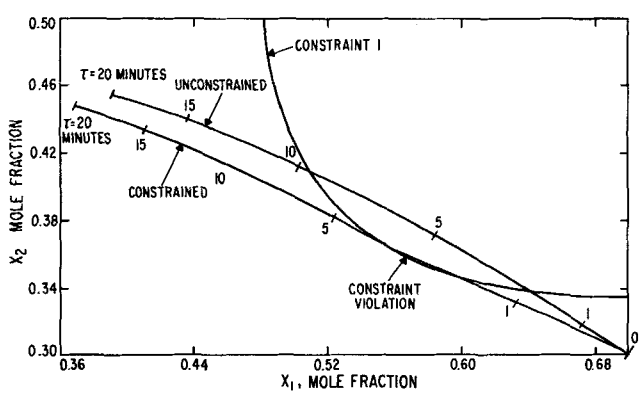

Fig. 1. Trajectory in state space for constraint 1.

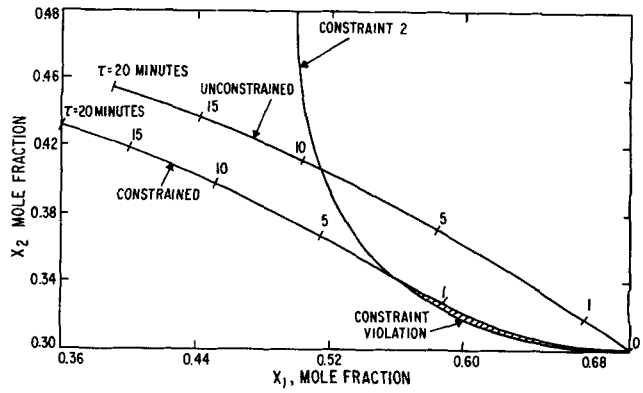

Fig. 2. Trajectory in state space for constraint 2.

$$T(\tau) = \frac{(E_1 - E_2)/R}{\ln\left\{\dfrac{E_1 G_1 x_1(\tau)\,[z_2(\tau) - z_1(\tau)]}{E_2 G_2 x_2(\tau)\, z_2(\tau)}\right\}}. \tag{29}$$

With the use of Equation (29) T can be eliminated from Equations (20), (21), (25), and (26). Following quasilinearization the state and adjoint equations are linearized. The procedure is straightforward, and the linearized equations will not be detailed here.

The reaction rate parameters used in this study are

$E_1 = 20{,}000$ cal./mole $\quad G_1 = 5.35 \times 10^{13}$ min.$^{-1}$
$E_2 = 28{,}000$ cal./mole $\quad G_2 = 4.61 \times 10^{19}$ min.$^{-1}$

where the reaction rate constant is given by Equation (23). The initial conditions were $x_1^o = 0.70$ and $x_2^o = 0.30$ with $\tau_f = 20$ min. Two constraints were considered, and the corresponding parameters for Equation (24) are given below:

$\eta_1 = -1.00$			$\eta_1 = -1.00$	
$\eta_2 = 0.69$			$\eta_2 = 0.80$	
$\eta_3 = 2.00$	con-		$\eta_3 = 2.00$	con-
$\eta_4 = 0.11$	straint		$\eta_4 = 0$	straint
$\eta_5 = -1.00$	1		$\eta_5 = -1.00$	2
$\eta_6 = -0.387025$			$\eta_6 = -0.40$	

No physical significance will be attached to these constraints or the form of $g(\mathbf{x})$ in Equation (24). The constraints were chosen merely for convenience and their ability to demonstrate the computational algorithm.

In the numerical scheme described above ϵ was set at 0.001, which represents better than 1% accuracy in $\mathbf{x}$ and $\mathbf{z}$. A fourth-order Runge-Kutta-Gill numerical integration routine was employed with a step size of 0.2. The algorithm was programmed for an IBM-7094 computer.

Typical results are presented in Figures 1 and 2 and Table 1. The unconstrained optimal trajectories are shown for comparison. The constrained trajectory shown in Figure 1 required a total of twenty-seven changes in z_c to reduce the constraint violation $x_3(\tau_f)$ from 5.035×10^{-4} for the unconstrained trajectory to 2.397×10^{-9} for the constrained trajectory. The trajectory plotted in Figure 2 required 11 z_c shifts to reduce $x_3(\tau_f)$ from 2.701×10^{-3} to 8.807×10^{-6}. The sequence of z_c values and the corresponding sequence of $x_3(\tau_f)$ values are plotted in Figure 3. In Table 1 note the effect of state constraints on the adjoint variable trajectories, particularly at $\tau = 0$. The difference in $\mathbf{z}(\tau)$ between the constrained and unconstrained state variable cases is much greater than the change in $\mathbf{x}(\tau)$ shown in Figures 1 and 2. A more detailed analysis of the results of this investigation follows.

CONVERGENCE

There are several factors which affect convergence and hence computation time. Some of these are the value of K

Table 1. Adjoint Trajectories and Optimal Control Policy for the Unconstrained and Constrained Systems

Unconstrained

τ	$z_1(\tau)$	$z_2(\tau)$	$T(\tau)$
0	−0.3599	−0.6736	289.2
4	−0.3040	−0.7516	287.5
8	−0.2401	−0.8197	286.4
12	−0.1682	−0.8826	285.6
16	−0.0882	−0.9423	285.0
20	0	−1.0000	284.5

Constraint 1

τ	$z_1(\tau)$	$z_2(\tau)$	$T(\tau)$
0	−0.0423	−0.1410	297.9
4	−0.2804	−0.7884	286.1
8	−0.2201	−0.8452	285.3
12	−0.1534	−0.8987	284.6
16	−0.0801	−0.9501	284.1
20	0	−1.0000	283.7

Constraint 2

τ	$z_1(\tau)$	$z_2(\tau)$	$T(\tau)$
0	+0.0045	−0.0135	312.9
4	−0.2834	−0.7839	286.3
8	−0.2227	−0.8420	285.4
12	−0.1553	−0.8967	284.8
16	−0.0811	−0.9491	284.2
20	0	−1.0000	283.8

Constraint 1: $-x_1^2 + 0.69\,x_1 + 2x_1x_2 + 0.11\,x_2 - x_2^2 - 0.387025 \leq 0$

Constraint 2: $-x_1^2 + 0.8x_1 + 2x_1x_2 - x_2^2 - 0.4 \leq 0$

in Equation (19), the desired tolerance ϵ_c, and the severity of the constraint.

To begin, consider the first factor. Solution of the tubular reactor problem with constraint 1 was attempted for various values of K. The z_c and $x_3(\tau_f)$ sequences are tabulated in Table 2. For $K = 1{,}000$ and $K = 10^4$ an ϵ_c of 10^{-6} was satisfied in twelve shifts of z_c. Both programs required about 11 min. of IBM-7094 computation time to achieve this. For both $K = 10$ and $K = 100$ excessive iteration was encountered at $m = 7$, 11, and 15. If quasilinearization could not satisfy $\epsilon = 10^{-3}$ for a given $z_c^{(m)}$ after ten trials, an excessive iteration condition was declared and iteration restarted with $z_c^{(m+1)} = 0.75 z_c^{(m)}$. In most cases about three iterations were required to meet $\epsilon = 10^{-3}$, and each iteration used about 10 sec. of computing time. It is difficult to say why excessive iteration occurs for $K = 10$ and not for $K = 1{,}000$. At iteration number 7, for example, the percentage change in z_c, defined as $[z_c^{(7)} - z_c^{(6)}] \times 100/z_c^{(6)}$, was 44.06% for $K = 10$, 44.16% for $K = 100$, and 44.02% for $K = 1{,}000$. Since these are essentially identical it is impossible to argue that failure to satisfy $\epsilon = 10^{-3}$ for $K = 10$ and 100 at $m = 7$ was due to too large an increase in z_c over the previous value. The excessive iteration is probably best viewed as an indicator of anomalous behavior. For $K = 10^5$, on the other hand, $z_c^{(1)}$ calculated from Equation (19) was definitely too large, and the quasilinearization algorithm failed at $m = 1$. In summary, a region of suitable K values appears to exist. Within this region convergence to $\epsilon_c = 10^{-6}$ was rapid. For K values less than the minimum suitable K, the excessive iteration condition existed, and this increased the computation time by increasing the number of iterations. For K greater than the maximum suitable K, convergence was completely unattainable.

Convergence rate is also a function of ϵ_c, as expected. It has already been seen that convergence was obtained with 12 z_c shifts for $K = 10^4$ and $\epsilon_c = 10^{-6}$ with constraint 1. However, in continuing the iteration of this case to $x_3(\tau_f) = 2 \times 10^{-9}$, the excessive iteration condition was encountered at $m = 13$, 17, and 22. Again no reason can be given for this behavior, although it is interesting to note that the excessive iteration condition occurs more frequently for $x_3(\tau_f) < 10^{-6}$ than for $x_3(\tau_f) > 10^{-6}$.

For constraint 1 it appears that $x_3(\tau_f)$ can be made as small as possible, although the excessive iteration condition was encountered as $x_3(\tau_f)$ became small. For the more severe constraint 2 this was not the case however. After 11 z_c shifts with $\epsilon = 10^{-3}$ and $K = 100$, $x_3(\tau_f)$ was reduced to less than 9×10^{-6}. The corresponding z_c was 163. It was impossible to reduce the penetration of constraint 2 any further. Even $z_c = 170$ resulted in excessive iteration. This indicates that the minimum attainable tol-

Table 2. The Effect of K on Convergence for Constraint 1

	$K = 10$		$K = 100$	
m	$z_c^{(m)}$	$x_3^{(m)}(\tau_f) \times 10^3$	$z_c^{(m)}$	$x_3^{(m)}(\tau_f) \times 10^3$
0	0	0.50353	0	0.50353
1	0.00504	0.50321	0.0504	0.50025
2	7.926	0.22024	7.728	0.22402
3	14.091	0.13920	13.954	0.14048
4	24.681	0.07748	24.425	0.07847
5	37.972	0.04500	37.673	0.04500
6	56.392	0.02584	55.957	0.02615
7	81.236	E	80.665	E
8	60.927	0.02299	60.500	0.02324
9	97.490	0.01115	96.708	0.01129
10	131.892	0.00684	130.928	0.00692
11	186.559	E	185.179	E
12	139.919	0.00626	138.884	0.00635
13	227.091	0.00276	226.299	0.00277
14	295.677	0.00177	294.239	0.00178
15	418.314	E	416.645	E
16	313.736	0.00159	312.484	0.00160

	$K = 1{,}000$		$K = 10{,}000$	
m	$z_c^{(m)}$	$x_3^{(m)}(\tau_f) \times 10^3$	$z_c^{(m)}$	$x_3^{(m)}(\tau_f) \times 10^3$
0	0	0.50353	0	0.50353
1	0.5035	0.47203	5.035	0.28565
2	8.048	0.21797	11.637	0.16476
3	14.521	0.13535	20.634	0.09507
4	25.124	0.07582	32.908	0.05414
5	38.628	0.04393	49.143	0.03143
6	57.233	0.02527	71.611	0.01806
7	82.420	0.01463	101.957	0.01038
8	117.061	0.00832	142.922	0.00603
9	162.766	0.00433	199.898	0.00345
10	266.078	0.00278	275.711	0.00201
11	311.783	0.00161	381.406	0.00113
12	429.535	0.00094	517.624	0.00067
13			713.317	E
14			534.988	0.00063
15			820.346	0.00029
16			1,064.18	0.00018
27			10,182.3	0.00002

	$K = 100{,}000$	
m	$z_c^{(m)}$	$x_3^{(m)}(\tau_f) \times 10^3$
0	0	0.50353
1	50.353	B

E = Excessive iteration. If convergence was not obtained within ten iterations for a given m, z_c is decreased by ¼ and iteration restarted.

B = Procedure blows up. Computation is terminated.

erance depends on the severity of the constraint, the third factor affecting convergence.

Figure 3 shows that the convergence rate is independent of the constraint severity. The slopes of the two curves in Figure 3 are nearly identical. Thus the ratio $x_3^{(m)}(\tau_f)/x_3^{o}(\tau_f)$ is nearly independent of the constraint severity. This is a very welcome property of the quasilinearization algorithm for systems with state constraints.

Before we leave Figure 3 it should be noted that the points plotted with circles and squares represent the actual sequence of z_c shifts for the indicated cases. While the sequences for other runs are not shown in Figure 3, it was found that the sequence pair $\{z_c^{(i)}\}$ and $\{x_3^{(i)}(\tau_f)\}$ is unique; that is, for each z_c there exists one and only one $x_3(\tau_f)$. This property, while of no real value to the numerical algorithm, does make the use of the quasilinearization scheme more satisfying from a mathematical point of view.

Constraints 1 and 2, in addition to varying in severity, produce two different types of state-constrained solutions. In Figure 1 note that $\mathbf{x}(0)$ lies off the boundary of constraint 1 and the constrained trajectory does not remain on the boundary but is merely tangent to it. On the other hand, the constrained trajectory for constraint 2, shown in Figure 2, starts on the boundary and appears to remain on it for a period of time. The latter is the usual case encountered in practice. A good test of the quasilinearization algorithm is to see how close convergence is when the trajectory follows the boundary. If $\mathbf{x}(\tau)$ lies on $g(\mathbf{x}) = 0$ for a finite period of time, then $\dot{g}(\mathbf{x}) = 0$. Thus it is possible to calculate x_2 for a given x_1 from $g(\mathbf{x}) = 0$ and the corresponding control T from $\dot{g}(\mathbf{x}) = 0$. This trajectory,

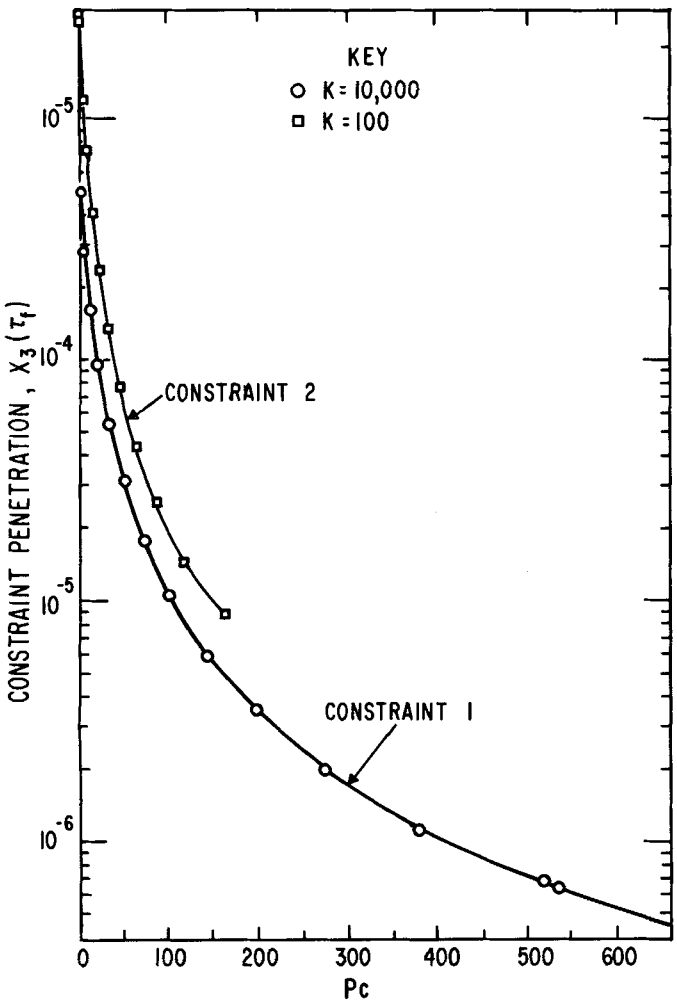

Fig. 3. Sequence of values and the corresponding constraint penetration sequence for constraints 1 and 2.

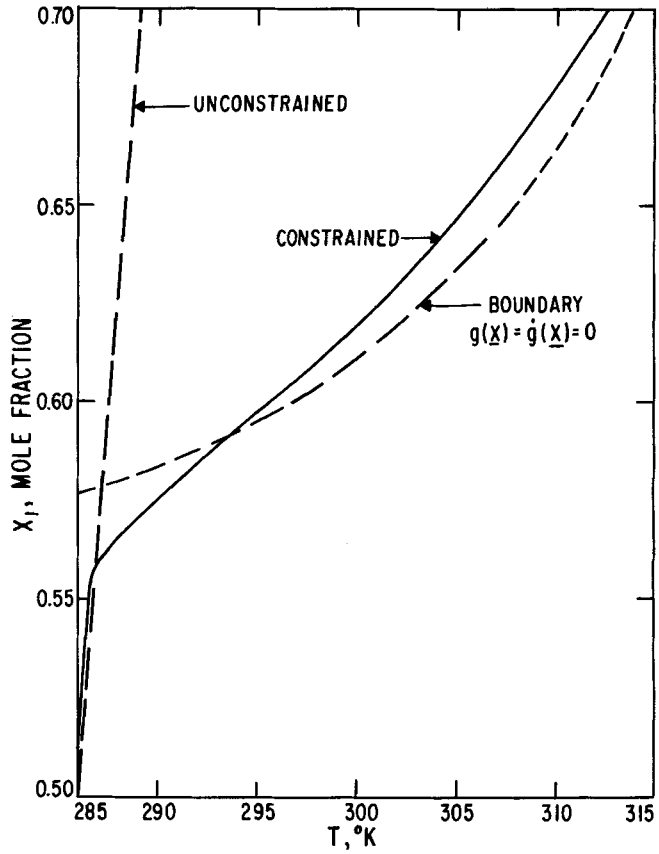

Fig. 4. The boundary segment of constraint 2.

the boundary-segment trajectory, is shown in Figure 4. Also plotted in Figure 4 are the unconstrained trajectory and the constrained trajectory for $z_c = 163$. The latter was obtained with $K = 100$ after 11 z_c shifts and is the one referred to previously in Figure 2 and Table 1. Note that the constrained T differs from the boundary-segment T by less than 2.5°K., a less than 1% discrepancy. This is small compared with the 24°K. difference between the unconstrained T and the constrained T at $\tau = 0$. It can therefore be concluded that the quasilinearization algorithm for state constraints satisfactorily treats cases where a portion of the trajectory follows the constraint boundary.

In summary, the numerical solution of the Amundson-Bilous tubular reactor problem with a quadratic state constraint yielded the following results:

1. The difference between constrained and unconstrained trajectories was greatest in $\mathbf{z}(\tau)$.
2. Quasilinearization reduced the constraint penetration by a factor of about 10^5 in 27 z_c iterations for the less severe constraint 1 and by a factor of about 10^3 in 11 z_c shifts for the more severe constraint 2.
3. The IBM-7094 computation time averaged about 30 sec./z_c shift using 100 numerical integration steps.
4. A definite region of satisfactory K values existed. Outside of this region either excessive iteration occurred or the algorithm blew up.
5. Difficulty was encountered in making the constraint penetration arbitrarily small. For constraint 2 it was not possible to reduce $x_3(\tau_f)$ below 9×10^{-6}. Although the penetration was reduced to about 10^{-9} for constraint 1, below 10^{-6} excessive iteration conditions occurred three times using $K = 10^4$.
6. The rate of convergence of the algorithm was independent of the severity of the constraint.
7. For each $x_c^{(m)}$ there corresponds a unique $x_3^{(m)}(\tau_f)$.
8. For constraint 2 part of the trajectory lies on the constraint boundary. Quasilinearization resulted in a con-

trol policy $T^*(\tau)$ which differed by less than 2.5°K. from the control required for the trajectory to remain on the boundary. This represents an error in $T^*(\tau)$ of less than 1%. The deviation between $T^*(0)$ for the unconstrained case and $T^*(0)$ for the constrained case was 24°K.

CONCLUSIONS

It has already been mentioned that the quasilinearization penalty function algorithm inherits all of the facets of the quasilinear method for unconstrained systems. Thus the method is hampered by drawbacks such as adjoint variable instability and poor initial guesses. By contrast, the desirable feature of quadratic convergence is maintained, suggesting that quasilinearization is theoretically superior to gradient methods for constrained state problems. In addition to these points, the investigation detailed here produces the following conclusions.

1. The sequence of penalty weighting functions $\{z_c^{(m)}\}$ generated by the application of a Newton-Raphson scheme is satisfactory in all cases.

2. Close tolerances are not easily obtained, especially when a portion of the trajectory lies on the constraint boundary. However, in the numerical example treated here attainable tolerances and the resulting control policies were acceptable in all cases.

3. The computation time required for convergence varied between 6 and 25 min. depending on the desired tolerance ϵ_c. Since the system treated here only contained two state variables, computation time may be excessive for large systems.

4. The constant K in Equation (19) must be determined by intuition, but a region of feasible K's definitely exists and it is sufficiently large to eliminate much trial and error in searching for K.

In summary, the results indicate that the quasilinearization algorithm with the incorporation of state constraints by penalty functions can effectively solve optimization problems subject to inequality constraints on the state of the system. In addition to the drawbacks inherent in quasilinearization, the use of penalty functions requires generating a sequence of solutions which in turn means more computation time that that needed for the solution of unconstrained systems. However, owing to the quadratic nature of quasilinearization convergence under certain conditions and its rapid convergence near the optimum, this algorithm may be more successful than the direct method of Denham and Bryson (8), which employs the necessary conditions for an optimum and the method of gradients. The ease of programming the penalty function method over the direct method is another consideration in favor of the algorithm investigated here. Thus, quasilinearization with penalty functions holds excellent promise of becoming a workable algorithm for solving this type of problem.

ACKNOWLEDGMENT

Financial assistance in the form of a Ford Foundation Fellowship, a Standard Oil Company of California Fellowship, and a National Science Foundation Grant NSF GK-460 is gratefully acknowledged. Use was made of Princeton University's computer facilities, supported in part by National Science Foundation Grant NSF GP-579.

NOTATION

B = function defined by Equation (8)
E_1, E_2 = activation energy, cal./ mole
E = tolerance
F_i = function defined by Equation (1) of reference *18*
g = state constraint
G_1, G_2 = pre-exponential factors, min.$^{-1}$
h = Heaviside function defined by Equation (12)
H = Hamiltonian
I = performance index
J = integral performance index
k_1, k_2 = rate constants defined by Equation (23), min.$^{-1}$
K = constant in Equation (19)
m = approximation number
n = dimension of state vector
r = dimension of control vector
t = independent variable
T = temperature, °K.
u_i = control variable
x_i = state variable
z_i = adjoint variable

Greek Letters

α_j = superposition constant
$\delta_i^{(l)}$ = Kronecker delta
∇ = gradient operator
ϵ = tolerance
η_i = constant in Equation (24)
μ = constant Lagrange multiplier in Equations (5) and (6)
ν = scalar function of t defined by Equation (4)
σ_i = penalty weighting coefficient
τ = tubular reactor residence time, minutes
ϕ = penalty function

Subscripts

c = constraint
f = final value
s = switching point

Superscripts

o = initial value
$*$ = optimal trajectory
$+$ = before the switching point
$-$ = after the switching point
$\sim$ = augmented functional

LITERATURE CITED

1. Aris, Rutherford, *Chem. Eng. Sci.*, **13**, 18 (1960).
2. Berkovitz, L. D., *J. Math. Anal. Appl.*, **5**, 488 (1962).
3. ———, and S. E. Dreyfus, *ibid.*, **10**, 275 (1965).
4. Bilous, Olegh, and N. R. Amundson, *Chem. Eng. Sci.*, **5**, 81, 115 (1956).
5. Bryson, A. E., and W. F. Denham, *J. Appl. Mech.*, **29**, No. 2, 247 (1962).
6. ———, and S. E. Dreyfus, *J. Am. Inst. Aeronaut. Astronaut*, **1**, 2544 (1963).
7. Chang, S. S. L., *Automatica*, **1**, 55 (1962).
8. Denham, W. F., and A. E. Bryson, *J. Am. Inst. Aeronaut. Astronaut*, **2**, 25 (1964).
9. Denn, M. M., and Rutherford Aris, *Ind. Eng. Chem. Fundamentals* **4**, 7 (1965).
10. Dreyfus, S. E., *J. Math. Anal. Appl.*, **4**, 297 (1962).
11. Guinn, T., *J. SIAM Control: Series A*, **3**, 181 (1965).
12. Kalaba, R., *J. Math. Mech.*, **8**, 519 (1959).
13. Kelley, H. J., "Optimization Techniques with Applications to Aerospace Systems," G. Leitmann, ed., p. 206, Academic Press, New York (1962).
14. Kopp, R. E., and R. McGill, "Computing Methods in Optimization Problems," A. V. Balakrishnan and L. W. Neustadt, ed., p. 65, Academic Press, New York (1964).
15. McGill, R., *J. SIAM Control: Series A*, **3**, 291 (1965).
16. Okamura, K., *ibid.*, **2**, 317 (1965).
17. Pontryagin, L. S., V. G. Boltyanskii, R. V. Gamkrelidze, and E. F. Mischenko, "The Mathematical Theory of Optimal Processes," Interscience, New York (1962).
18. Rothenberger, B. F., and Leon Lapidus, *AICHE J.*, **13**, 973 (1967).
19. Rozonoer, L., *Automation and Remote Control*, **20**, 1288 1405, 1517 (1960).
20. Russell, D. L., *J. SIAM Control: Series A*, **2**, 409 (1965).

Manuscript received November 9, 1966; revision received January 19, 1967; paper accepted January 23, 1967. Paper presented at AIChE Houston meeting.