Technical Notes

TECHNICAL NOTES are short manuscripts describing new developments or important results of a preliminary nature. These Notes cannot exceed 6 manuscript pages and 3 figures; a page of text may be substituted for a figure and vice versa. After informal review by the editors, they may be published within a few months of the date of receipt. Style requirements are the same as for regular contributions (see inside back cover).

Use of Quasi-Linearization in the Computation of Optimal Singular Controls

B. P. Yeo*
University of Singapore, Singapore

I. Introduction

In recent years, there has been considerable interest in the formulation of numerical techniques to solve optimal control problems with singular arcs. It is because these arcs appear in a large number of problems of practical interest.[1-5] The numerical methods which are currently available to solve problems of this type without discretizing the state equations and the performance index are the gradient method, the method of Anderson,[6] the method of Jacobson et al.,[7] and the modified quasilinearization method.[8] The gradient method is generally slow to converge to singular arcs. In applying the method of Anderson,[6] the sequence of controls for the problem must be known. The modified quasilinearization method[8] is only applicable to the computation of totally singular optimal control problems. Edgar and Lapidus[3] noted that it is almost impossible to establish the existence of optimal singular arcs without the actual numerical solution of a given control problem. In the ε algorithm of Jacobson et al.,[7] the singular control is made nonsingular by the addition of an integral quadratic function of the control variable to the performance index. A parameter $\varepsilon > 0$ multiplies this added functional. The resulting nonsingular problem is solved for a monotonically decreasing sequence (ε; $\varepsilon_1 > \varepsilon_2 > \cdots > \varepsilon_k > 0$). As $k \to \infty$ and $\varepsilon_k \to 0$, the solution of the modified problem tends to the solution of the original singular problem. Jacobson et al.[7] noted that as $k \to \infty$, numerical difficulties may arise when using differential dynamic programing to solve the modified problem. A variant of the ε algorithm to overcome these difficulties, known as the $\varepsilon-\alpha$ algorithm, is also given in Ref. 7. Crude solutions of four linear problems using the ε algorithm are presented in Ref. 7. Refinement of these crude solutions to the desired accuracy is achieved by the application of the $\varepsilon-\alpha$ algorithm. Jacobson et al.[7] noted that the results may be optimistic as the examples solved are very simple and none have the characteristics of long sensitive trajectories. Furthermore, it is necessary to use penalty functions to treat fixed terminal states.

In this paper, the singular problem is made nonsingular as the $\varepsilon-\alpha$ algorithm of Jacobson et al.[7] A quasilinearization algorithm is then used to solve the sequence of modified problems. Details of the quasilinearization algorithm are presented in Sec. 2–4. In applying the proposed method, no penalty function is necessary to treat fixed terminal states. Two problems were solved to illustrate the usefulness of the proposed method. The effect of the parameter ε on the convergence characteristics of the quasilinearization method is investigated. Within the range $\varepsilon \leqq 1$, it is shown that the larger the value of ε, the greater is the likelihood of convergence and the slower is the rate of convergence.

Received February 14, 1974; revision received July 15, 1974.
Index category: Navigation, Control, and Guidance Theory.

* Lecturer, Department of Industrial and Systems Engineering, Faculty of Engineering.

II. Statement of the Singular Problem

Consider a dynamic system described by

$$\dot{x} = f_1(x,t) + f_2(x,t)u$$
$$x(0) = x_0, \quad x(t_f) = x_f \tag{1}$$

Here the symbol x, an n-dimensional column vector, denotes the state; the symbol u, an m-dimensional column vector, denotes the control; f_1 is an n-dimensional column vector function; f_2 is an $n \times m$ matrix. The control is constrained in the following way:

$$u_{\min} \leqq u(t) \leqq u_{\max} \tag{2}$$

where $u_{\max}$ and $u_{\min}$ are m-dimensional column vectors denoting the upper and lower bounds of the control vector respectively.

The object is to minimize the performance index

$$J = \int_0^{t_f} [L_1(x,t) + L_2^T(x,t)u]\, dt \tag{3}$$

where the final time t_f is given explicitly. In Eq. (3) L_1 is a scaler function and L_2 is an m-dimensional column vector function. The superscript T denotes the transpose of a matrix. It is assumed that L_1, L_2, f_1, f_2 have continuous first and second derivatives relative to (x,t). It is also assumed that the optimal solution exists.

III. Modified Problem

The modified problem is obtained by adding a quadratic function of the control vector to the performance index of the singular problem. The resulting problem is to minimize the modified performance index.

$$J' = \int_0^{t_f} \left[L_1(x,t) + L_2^T(x,t)u + \frac{\varepsilon}{2}(u-\alpha^s)^T(u-\alpha^s) \right] dt \tag{4}$$

subject to the constraints in Eq. (1). In the previous expression, the parameter ε is a small, preselected positive quantity; $\alpha^s(t)$ is an m-dimensional column vector function. The superscript s denotes the sth modified problem (initially, $s = 1$).

The modified problem is a problem of the class of nonsingular optimal control problem of Ref. 10. The solution of this class of problem can be generated using the quasilinearization algorithms of Ref. 10, or Refs. 11 and 12.

IV. Description of the Quasi-Linearization Algorithm

A step-by-step description of the algorithm is outlined below.

a) Choose the parameter ε, the nominal functions $x^1(t)$, $u^1(t)$, $\lambda^1(t)$, and set $\alpha^1(t) = u^1(t)$.

b) Solve the modified problem using the quasilinearization algorithm of Ref. 10. Convergence is achieved when $\rho < \delta$, a small, preselected positive quantity. This yields $x^s(t)$, $u^s(t)$, $\lambda^s(t)$ (initially, $s = 1$, $N = 0$, where N is the number of iterations).

c) Stop computation when

$$P = Q_1/Q_2 < \xi \tag{5}$$

where

$$Q_1 = \int_0^{t_f} \frac{\varepsilon}{2}(u-\alpha^s)^T(u-\alpha^s)\, dt \tag{6}$$

$$Q_2 = \int_0^{t_f} \left[L_1(x,t) + L_2^T(x,t)u + \frac{\varepsilon}{2}(u-\alpha^s)^T(u-\alpha^s) \right] dt \tag{7}$$

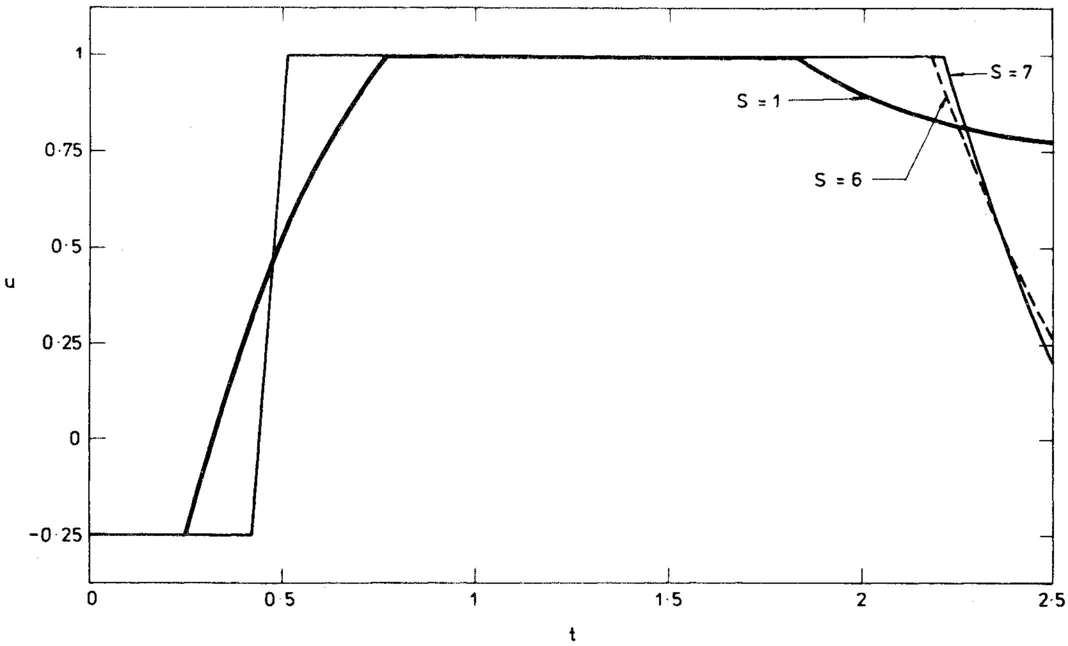

Fig. 1 Converged controls at $s = 1, 6, 7$ vs t, $\varepsilon = 1$.

and ξ is a small, preselected positive quantity. Otherwise, set $x^{s+1}(t) = x^s(t)$, $u^{s+1}(t) = u^s(t)$, $\lambda^{s+1}(t) = \lambda^s(t)$, $\alpha^{s+1}(t) = u^{s+1}(t)$, $s = s+1$, $N = N_s^*$, where N_s^* is the number of iterations at convergence to the solution of the sth modified problem. Go to b).

In Ref. 7, Jacobson et al. proved that as $Q_1 \to 0$ (and therefore $P \to 0$), the solution of the modified problem approaches the singular solution of the original problem.

V. Numerical Examples

In order to study the usefulness and convergence characteristics of the proposed method, two numerical examples were developed using an IBM 360/50 digital computer and double precision arithmetic. Computer runs were made on each example with different values of the parameter ε. After selecting the initial nominal functions, the problem was solved using the quasi-linearization algorithm of Ref. 10. The interval of integration was divided into 100 steps and a fourth-order Runge-Kutta procedure was used to integrate first-order linear differential equations. Sets of linear algebraic equations were solved using the Gauss-Jordon method.

Convergence to the solution of each modified problem was defined as having occurred when

$$\rho < 10^{-6} \tag{8}$$

where

$$\rho = \sum_{i=1}^{n} \max_{t\varepsilon[0,t_f]} |x_{i(N+1)} - x_{i(N)}| + \sum_{i=1}^{n} \max_{t\varepsilon[0,t_f]} |\lambda_{i(N+1)} - \lambda_{i(N)}| + \sum_{j=1}^{m} \max_{t\varepsilon[0,t_f]} |u_{j(N+1)} - u_{j(N)}| \tag{9}$$

Convergence of the proposed quasi-linearization algorithm was defined as follows:

$$P < 10^{-4} \tag{10}$$

Example 1

Choose $u(t)$ to minimize

$$J = \int_0^{2.5} [x_1^2 + x_2^2 + u]\, dt \tag{11}$$

subject to the constraints

$$\dot{x}_1 = x_2, \quad x_1(0) = 1, \quad x_1(2.5) = 0 \tag{12}$$

$$\dot{x}_2 = (1 - x_1^2)x_2 - x_1 + u, \quad x_2(0) = 0, \quad x_2(2.5) = 0 \tag{13}$$

and $-0.25 \leq u \leq 1$.

The following initial nominal functions were used to start the proposed quasi-linearization algorithm.

$$x_1(t) = 1 - t/2.5, \quad x_2(t) = 0, \quad \lambda_1(t) = 0, \quad \lambda_2(t) = -1,$$
$$u(t) = -0.25 + 0.5t$$

Computer runs were made with the parameter ε in the range

$$0.001 \leqq \varepsilon \leqq 1 \tag{14}$$

Table 1 shows the number of iterations at convergence N^*, the number of modified problems at convergence s^*, vs the parameter ε. Obviously, the proposed method diverges at smaller values of ε. Furthermore, the smaller the value of the parameter ε, the quicker is the rate of convergence if the algorithm does converge. For $\varepsilon = 1$, the converged controls of the modified problems at $s = 1, 6, 7$ are displayed in Fig. 1.

Example 2

Choose $u(t)$ to minimize

$$J = \int_0^1 x_2^2\, dt \tag{15}$$

and subject to the constraints

$$\begin{aligned} \dot{x}_1 &= x_2, \quad x_1(0) = 0, \quad x_1(1) = 1 \\ \dot{x}_2 &= 8.55u - 2.406x_2, \quad x_2(0) = 0, \quad x_2(1) = 0 \end{aligned} \tag{16}$$

and

$$-1 \leqq u \leqq 1 \tag{17}$$

Table 1 Convergence characteristics of Example 1

ε	s^*	N^*
1	7	25
0.5	6	23
0.1	3	16
0.075	Divergence	Divergence
0.05	Divergence	Divergence
0.01	Divergence	Divergence
0.005	Divergence	Divergence
0.001	Divergence	Divergence

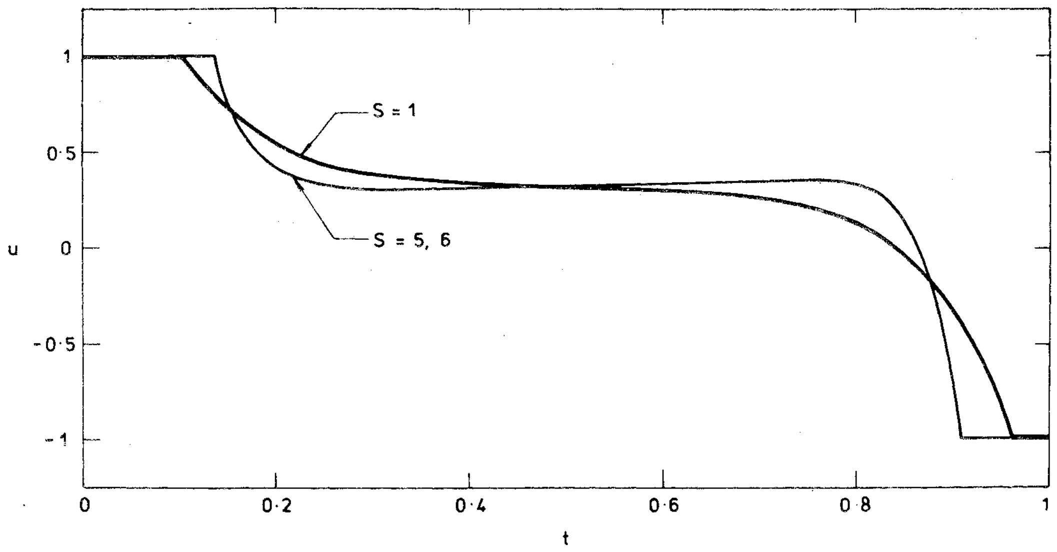

Fig. 2 Converged controls at $s = 1, 5, 6$ vs t, $\varepsilon = 1$.

The following initial nominal functions were used to start the proposed quasi-linearization algorithm.

$$x_1(t) = t, \quad x_2(t) = 0, \quad \lambda_1 = 0, \quad \lambda_2(t) = 0, \quad u(t) = 1 - 2t.$$

Computer runs were made with the parameter ε in the range

$$0.001 \leqq \varepsilon \leqq 1 \tag{18}$$

Table 2 shows the number of iterations at convergence N^*, the number of modified problems at convergence s^*, vs the parameter ε. Obviously, the proposed method diverges at smaller values of ε. Furthermore, the smaller the value of the parameter ε, the quicker is the rate of convergence if the algorithm does converge. For $\varepsilon = 1$, the converged controls of the modified problems at $s = 1, 5, 6$ are displayed in Fig. 2.

VI. Conclusion

The investigation detailed in this paper produces the following conclusions.

1) When it converges, the quasi-linearization algorithm appears to offer simpler programing and faster computational speed compared to differential dynamic programing. Furthermore, no penalty function is needed to treat fixed terminal states.

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

3) Within the range $\varepsilon \leqq 1$, the smaller the value of ε, the smaller is the likelihood of convergence and the quicker is the rate of convergence if the quasi-linearization algorithm does converge.

4) With crude initial nominal functions, the proposed quasi-linearization method converges rapidly to the singular solution within the range of feasible ε's.

Table 2 Convergence characteristics of Example 2

ε	s^*	N^*
1	6	18
0.75	5	16
0.5	4	15
0.25	3	12
0.1	Divergence	Divergence
0.05	Divergence	Divergence
0.01	Divergence	Divergence
0.001	Divergence	Divergence

5) There is no need to guess the sequence of controls.

A major problem of the quasi-linearization method is the choice of initial nominal functions. In general, the degree of difficulty in locating the bounded region of convergence increases with the dimensions of the system and the control. In a real system, the initial selection of the state and control variables can be made by reference to physical considerations. Concerning the adjoint variables, the method of Yeo et al.[10] can be used to choose them optimally. This, together with the advantages discussed earlier, makes the proposed method a possible workable algorithm for solving practical singular control problems of large dimensional systems.

References

[1] Anderson, G. M. and Watt, G. W., "Air-to-Air Missile Trajectories for Maximization of Launch Range," *Journal of Spacecraft and Rockets*, Vol. 8, No. 7, July 1971, pp. 747–751.

[2] Bryson, A. E. and Ho, Y. C., "Applied Optimal Control," Blaisdell, Waltham, Mass., 1969.

[3] Edgar, T. F. and Lapidus, L., "The Computation of Optimal Singular Bang-Bang Control, I: Linear Systems," *American Institute of Chemical Engineers Journal*, Vol. 18, No. 4, 1972, pp. 774–779.

[4] Edgar, T. F. and Lapidus, L., "The Computation of Optimal Singular Bang-Bang Control, II: Nonlinear Systems," *American Institute of Chemical Engineers Journal*, Vol. 13, No. 5, 1972, pp. 780–785.

[5] Power, W. F. and McDanell, J. P., "Switching Conditions and a Synthesis Technique for the Singular Guidance Problem," *Journal of Spacecraft and Rockets*, Vol. 8, No. 10, Oct. 1971, pp. 1027–1032.

[6] Anderson, G. M., "An Indirect Numerical Method for the Solution of a Class of Optimal Control Problems with Singular Arcs," *IEEE Transaction on Automatic Control*, Vol. AC-17, No. 3, June 1972, pp. 363–365.

[7] Jacobson, D. H., Gershwin, S. B., and Lele, M. M., "Computation of Optimal Singular Controls," *IEEE Transactions on Automatic Control*, Vol. AC-15, No. 1, Feb. 1970, pp. 67–73.

[8] Aly, G. M. and Chan, W. C., "Application of a Modified Quasi-linearization Technique to Totally Singular Optimal Control Problems," *International Journal of Control*, Vol. 17, No. 4, April 1973, pp. 809–815.

[9] Jacobson, D. M., "New Second-Order and First-Order Algorithms for Determining Optimal Control: A Differential Dynamic Programming Approach," *Journal of Optimization Theory and Applications*, Vol. 2, No. 6, 1968, pp. 411–440.

[10] Yeo, B. P., Waldron, K. J., and Goh, B. S., "Optimal Initial Choice of Multipliers in the Quasi-linearization Method for Optimal Control Problems with Bounded Controls," *International Journal of Control*, Vol. 20, No. 1, July 1974, pp. 17–33.

[11] Leondes, C. T. and Paine, G., "Extensions in Quasi-linearization Techniques for Optimal Control," *Journal of Optimization Theory and Applications*, Vol. 2, No. 5, 1968, pp. 316–330.

[12] Leondes, C. T. and Paine, G., "Computational Results for Extensions in Quasi-linearization Techniques for Optimal Control," *Journal of Optimization Theory and Applications*, Vol. 2, No. 6, 1968, pp. 395–410.

Mach Disk Location in Jets in Co-Flowing Airstreams

FRANK I. BUCKLEY JR.*
University of Maryland, College Park, Md.

Nomenclature

M_j = Mach number at nozzle exit
M_∞ = freestream Mach number
P_j = static pressure at nozzle exit
P_∞ = freestream static pressure
r_b = base radius of body
r_j = nozzle exit radius
x_s = axial location of Mach disk measured downstream from nozzle exit plane
γ = ratio of specific heats
θ_n = nozzle exit angle

Introduction

SYSTEM design requirements frequently necessitate a prediction of the structure of the plume formed by the exhaust of an underexpanded nozzle flow into the surrounding environment. The difficulty of this prediction is a function of the degree of underexpansion involved, the downstream distance over which the plume is to be defined, and the motion of the environment relative to the plume. A condition of particular interest in flight situations where exhaust plume signal interference and/or infrared signature must be assessed, involves the exhaust of a moderately under-expanded flow into a co-flowing airstream. In such cases the structure of a plume must be defined in regions dominated by gas dynamic effects in the vicinity of the nozzle exit plane, and by the effects of turbulent mixing, and frequently chemistry, further downstream. Although no practical means are presently available for the general computation of such complex flowfields, efforts continue in the development of flow models which attempt to account for the influence of those effects considered to have a major influence on the plume structure.[1–4] This Note concerns one particular aspect of the prediction problem, namely, the axial location of the Mach disk in an underexpanded jet in a co-flowing airstream.

Disk Location for Exhaust into a Still Environment

The foundation of the present work lies in a review of past research concerning the Mach disk location in plumes formed following the exhaust of an underexpanded flow into a still environment. Reporting on an extensive series of experimental and theoretical studies, Love et al.[5] found that the axial location of the Mach disk is not strongly affected by variations in nozzle exit angle. For a given ratio of specific heats, then, the axial Mach disk location (nondimensionalized in terms of the nozzle exit radius) could be determined by specification of the exit-to-ambient pressure ratio and the exit Mach number.

Received March 24, 1974.
Index categories: Jets, Wakes, and Viscid-Inviscid Flow Interactions; Shock Waves and Detonations; LV/M Propulsion System Integration.
* Associate Professor, Department of Mechanical Engineering. Member AIAA.

A later study by Adamson and Nicholls[6] confirmed the findings of Love et al., but more importantly, resulted in a method for the analytical prediction of the Mach disk location which was found to be in substantial agreement with the experiment. In summary, they hypothesized that the Mach disk would be found at a point along the nozzle axis where the compression of the expanded plume flow through a normal shock wave resulted in the elevation of the local static pressure to the ambient value. To implement this hypothesis, they noted that the plume flowfield in the region bounded by the nozzle exit plane, the intercepting shock wave, and the Mach disk, is for a given ratio of specific heats, solely a function of the flow conditions at the nozzle exit plane.[7] Thus, they were able to utilize the results of an earlier study for the flowfield produced by exhaust from a near sonic orifice into a vacuum[7] to predict the axial location of the Mach disk as a function of exit-to-ambient pressure ratio and exit Mach number for a ratio of specific heats of 1.4.

The experimental work of Lewis and Carlson[8] included a consideration of the effect of the ratio of specific heats on the Mach disk location, and, in general, greatly simplified the prediction procedure. They found that the axial locations of the Mach disk observed in their studies, as well as those that had been determined by earlier investigators, could be correlated by using the equation, $x_s/r_j = 1.38M_j(\gamma p_j/p_\infty)^{0.5}$.

Disk Location for Exhaust into a Co-Flowing Airstream

Consider, now, the situation in which the underexpanded flow exhausts into a co-flowing airstream. Schlieren photographs presented by Love et al.[5] for the case in which the external flow is supersonic reveal the absence of shock waves in the external flow downstream of the recompression shock located at the end of the first plume wavelength. This suggests, then, that the pressure in the plume, and in the external flow, is nearly ambient downstream of the first plume wavelength. Further, measurements by Furby[9] demonstrate that a similar condition prevails for exhaust into a still environment. Thus, it is reasonable

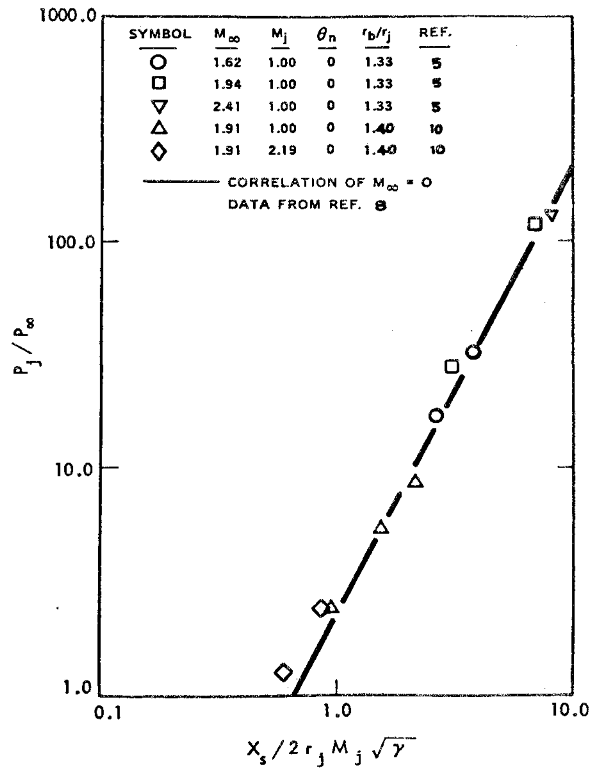

Fig. 1 Correlation of Mach disk location in plumes and in the presence of co-flowing streams.