KYBERNETIKA — VOLUME 11 (1975), NUMBER 3

An Approximative Method for Solving the Non-linear Optimal Control Problem

NGUYEN CANH, ANTONÍN TUZAR

This paper presents an algorithm for the computation of the optimal control problems, which can be reduced to the non-linear two-point boundary-value problem with the constraint on the control and with fixed final time. The algorithm is one of the indirect methods.

I. INTRODUCTION

Application of the maximum principle or the calculus of variations to optimal control problems, in general, reduces to a non-linear two-point boundary value problem. Therefore, for solving the non-linear optimal control problems there are two numerical approaches, the direct and indirect methods. The direct methods proceed by solving a sequence of non-optimal problems with the property that each successive set of solution functions yields an improved value for the functional being optimized. The indirect methods are concerned with finding, by numerical means, a set of functions that satisfy the necessary conditions, sometimes the sufficient conditions for an extremal.

In this paper we shall consider the control system defined by the vector differential equation with the boundary conditions

$$\dot{x} = f(x, u)\,, \quad x(t_0) = x_0\,, \quad x(t_f) = x_f\,, \tag{1}$$

where x is an n-dimensional vector, $x = (x^1, \ldots, x^n)$, which is continuous in the time t . $x(t)$ represents the physical state of the control system. $f(x, u)$ vector with coordinates $(f^1(x, u), \ldots, f^n(x, u))$ has first partial derivatives with respect to its all variables. The control $u(t)$ is a scalar function. Suppose further that the control function satisfies the inequality

$$|u(t)| \leqq 1\,. \tag{2}$$

The optimal control problem is to find a such control $u(t)$ in order to bring the system from a given initial state to a given terminal state so that to minimize functional

(3) $$J(u) = \int_{t_0}^{t_f} F(\boldsymbol{x}, u)\, \mathrm{d}t\,,$$

where t_f is fixed, and the integral is evaluated along the solution of (1) corresponding to the choise of $u(t)$. For an optimal control we obtain its corresponding trajectory $\boldsymbol{x}(t)$. The pair $(\boldsymbol{x}, u)$ is an optimal solution of the problem.

Applying the transformation of Valentine we may translate the inequality constraint to the equality constraint. Introducing the function α such that the equality

(4) $$1 - u^2(t) - \alpha^2(t) = 0$$

is satisfied on the interval $[t_0, t_f]$, then the problem above is reduced to the Lagrange problem.

II. THE NEWTON'S METHOD

Nowadays the generalized Newton's method is a powerful and commonly used numerical method for the solution of the non-linear equations. In particular, it is applied to systems of non-linear differential equations with two-point boundary conditions, and to the optimal control problems by solving the associated two-point boundary-value problems.

Consider the following non-linear operator equation

(5) $$P(x) = 0\,,$$

where $P(x)$ is a non-linear mapping from X to Y and X, Y are Banach-spaces. Suppose that P is differentiable in the Frechet sense i.e. at a fixed element $\boldsymbol{x}_0 \in X$ there exists a linear operator $A \in [X \to Y]$ such that

(6) $$\lim_{\|h\| \to 0} \frac{\|P(\boldsymbol{x}_0 + \boldsymbol{h}) - P(\boldsymbol{x}_0) - A(\boldsymbol{x}_0)\, \boldsymbol{h}\|}{\|\boldsymbol{h}\|} = 0\,,$$

for an arbitrary $\boldsymbol{x} \in X$. In this case the operator $A(\boldsymbol{x}_0)$ is the derivative of the operator P at the element $\boldsymbol{x}_0 \in X$ and we write

(7) $$A(\boldsymbol{x}_0) = P'(\boldsymbol{x}_0)\,.$$

Further suppose that $P'(\boldsymbol{x})$ is continuous in X, the operator P has zero in X, i.e., with the element $\boldsymbol{x} \in X$ we have

$$P(\boldsymbol{x}) = 0\,.$$

Starting with $x_0 \in X$ we replace $P(x)$ by

$$P(x_0) + P'(x_0)[x_1 - x_0] . \tag{8}$$

Putting this expression equal to zero, we solve the resulting linear equation for x_1, and so forth. Generally we have the following sequence of linear equations:

$$P(x_n) + P'(x_n)[x_{n+1} - x_n] = 0 , \quad n = 0, 1, 2, \ldots \tag{9}$$

Each x_n is an approximate solution of Eq. (5) and under approximative conditions the sequence $\{x_n\}$ converges to the solution of equation (5). In the other words we replace the original non-linear problem with a sequence of linear problems. The method setting the sequence $\{x_n\}$ as above is called the original Newton's method. P. Kenneth and G. E. Taylor used this method for solving the optimal control problem above.

If the sequence $\{x_n\}$ converges to the solution x^* and x_0 is collected sufficiently near x^*, then from the continuity of the operator P', the operators $P'(x_n)$ and $P'(x_0)$ are different only a little. Therefore we may replace $P'(x_n)$ with $P'(x_0)$. The sequence (9) becomes

$$P(x_n) + P'(x_0)[x_{n+1} - x_n] = 0 , \quad n = 0, 1, \ldots . \tag{10}$$

The method setting this sequence $\{x_n\}$ is called the modified Newton's method. On the other hand if applying the original Newton's method we have calculated the operator $P'(x_n)$ at each iteration. The main way of avoiding this disadvantage in the calculation process is to use the modified Newton's method. In addition to the modified Newton's method represented above in practice some computational effort may be saved. The gradient may be held constant after two or more iterations.

III. NUMERICAL WORK AND EXAMPLES

1. Numerical work

We well know that solving of the optimal control problems with inequality constrains for the control is reduced to solving of the control problems with the equality constraints. Solving this by the generalized Newton's method means solving the operator equations, which consist of the Euler-Lagrange equations, established the physical state differential equations and the equations, which represent the equality constraints for the control. It implies that the resulting non-linear system consists of two-point boundary-value problem of order $2n$ in addition to a system of scalar equations of order $2 + 1$. The generalized Newton's method is applied to this non-linear operator equation.

The method consists of in guessing a nominal control function, a nominal trajectory, a nominal multiplier function and a supplement function and then linearizing equations round the guessed functions. The linear two-point boundary-value problems are then solved, which yield corrections to the guessed functions.

For clarity of presentation we introduce

(11)
$$
\begin{aligned}
x^i(t) &= x^i(t)\,, && i = 1, 2, \ldots, n\,,\\
p^j(t) &= x^{n+j}(t)\,, && j = 1, 2, \ldots, n\,,\\
u(t) &= x^{2n+1}(t)\,,\\
\lambda(t) &= x^{2n+2}(t)\,,\\
\alpha(t) &= x^{2n+3}(t)\,.
\end{aligned}
$$

Setting the column vector

$$
x(t) = \begin{pmatrix} x^1(t) \\ \cdots\cdots \\ \cdots\cdots \\ x^{2n+3}(t) \end{pmatrix}
$$

then the Euler equations will be rewritten as

(12)
$$
\begin{aligned}
\dot{x}^i(t) &= f^i(x, t)\,, && i = 1, \ldots, n\,,\\
\dot{x}^j(t) &= f^j(t)\,, && j = n + 1, \ldots, 2n\,,\\
W^k(t) &= 0\,, && k = 2n + 1\,,\ \ldots,\ 2n + 3\,,
\end{aligned}
$$

with $2n$ boundary conditions for

(13)
$$
x^i(t_0) \quad \text{and} \quad x^i(t_f)\,, \quad i = 1, \ldots, n\,.
$$

The equation (12) may now be written in the vector form as follows

(14)
$$
\begin{gathered}
\dot{x}(t) = f(x, t)\,, \quad t \in [t_0, t_f]\,,\\
W(x, t) = 0\,.
\end{gathered}
$$

where

$$
\begin{aligned}
x &= (x^1, \ldots, x^{2n+3})^{\mathrm{T}}\,,\\
f &= (f^1, \ldots, f^{2n})^{\mathrm{T}}\,,\\
W &= (W^{2n+1}, \ldots, W^{2n+3})^{\mathrm{T}}\,.
\end{aligned}
$$

For the original Newton's method (9), the algorithm now requires the solution of the sequence of linear equations

$$\dot{x}_{n+1} = J(x_n, t)\,[x_{n+1} - x_n] + f(x_n, t) \tag{15}$$

with boundary conditions (13) and

$$0 = I(x_n, t)\,[x_{n+1} - x_n] + W(x_n, t)\,, \quad n = 0, 1, \dots\,; \tag{16}$$

n denotes the n-th iterate. Here

$$J(x_n, t) = \left[\frac{\partial f^i(x, t)}{\partial x^j}\right]_{x = x_n},$$

$$i = 1, \dots, 2n\,, \quad j = 1, \dots, 2n + 3\,,$$

and

$$I_{ij}(x_n, t) = \frac{\partial W^i}{\partial x^j}\bigg|_{x = x_n}$$

$$i = 2n + 1, \dots, 2n + 3\,, \quad j = 1, \dots, 2n + 3\,.$$

Quite similarly, for the modified Newton's method the algorithm now requires the solution of the sequence of linear equations:

$$\dot{x}_{n+1} = J(x_0, t)\,[x_{n+1} - x_n] + f(x_n, t)\,, \tag{17}$$

$$0 = I(x_0, t)\,[x_{n+1} - x_n] + W(x_n, t)\,, \tag{18}$$

$n = 0, 1, \dots,$ n denotes the n-th iterative.

Here $J(x_0, t)$ and $I(x_0, t)$ are the matrices with elements

$$J_{ij}(x_0, t) = \frac{\partial f^i(x, t)}{\partial x^j}\bigg|_{x = x_0},$$

$$i = 1, \dots, 2n\,, \quad j = 1, \dots, 2n + 3\,.$$

and

$$I_{ij}(x_0, t) = \frac{\partial W^i}{\partial x^j}\bigg|_{x = x_0},$$

$$i = 2n + 1, \dots, 2n + 3\,, \quad j = 1, \dots, 2n + 3\,.$$

At every iterative $n + 1$ the control function $u_{n+1}(t) = x^{2n+1}_{n+1}(t)$ is obtained from equation (16) for the original Newton's method or (18) for the modified Newton's method. The $u_{n+1}(t)$ is functional of the functions $x^1_{n+1}(t), \dots, x^{2n}_{n+1}(t)$. Therefore those relations are used to eliminate $u_{n+1}(t)$ from (15) or (17). The functions

$x_{n+1}^1(t), \ldots, x_{n+1}^{2n}(t)$ are then computed from (15) or (17), there is solving the special linear two-point boundary-value problem, using the complementary functions method [3]. Afterwards the control function $u(t)$ is computed from (16) or (18).

The iteration proceeds until

$$(19) \qquad \varrho(x_{n+1}, x_n) \leqq \delta ,$$

where

$$(20) \qquad \varrho(x_{n+1}, x_n) = \sum_{i=1}^{2n} \max_{t \in [t_0, t_f]} |x_{n+1}^i - x_n^i|$$

and δ is a suitably small positive constant. The corresponding iterative x_{n+1} is then accepted as the solution of the above optimal control problem.

Remark. If the control vector $\boldsymbol{u}$ is a r-dimensional vector, $\boldsymbol{u} = (u^1, \ldots, u^r)^T$ and exist r relations, which describe the constraints on $\boldsymbol{u}$, then we have

$$\lambda^i(t), \quad i = 2n + r + 1, \ldots, 2n + 2r ,$$

$$\alpha^j(t), \quad j = 2n + 2r + 1, \ldots, 2n + 3r ,$$

and

$$\boldsymbol{x} = (x^1, \ldots, x^{2n+3r})^T ,$$

$$\boldsymbol{w} = (w^{2n+1}, \ldots, w^{2n+3r})^T .$$

2. Illustrative examples

Example 1. We shall consider the non-linear control system defined by the Ricatti equation:

$$(21) \qquad \dot{x} = x^2 - \tfrac{3}{2}x + u$$

with the boundary conditions:

$$(22) \qquad x(0) = 1 , \quad x(1) = \tfrac{1}{2} .$$

In addition is given the inequality constraint

$$|u(t)| \leqq 1 \quad \text{for} \quad t \in [0, 1] .$$

Determine the state variable $x(t)$ and the control variable $u(t)$ so as to minimize

$$(23) \qquad J[u] = \int_0^1 u(t)\,\mathrm{d}t .$$

By introducing the function $\alpha(t)$ the inequality constraint on $u(t)$ may be rewritten as

$$(24) \qquad 1 - u^2 - \alpha^2 = 0 .$$

We introduce the multipliers $p(t)$ and $\lambda(t)$. The notation $\Phi(t)$ will be used for the function:

$$\Phi(t) = u + p(\dot{x} - x^2 + \tfrac{3}{2}x - u) + \lambda(1 - u^2 - \alpha^2).$$

The Euler equations are:

$$\dot{p} - p(\tfrac{3}{2} - 2x) = 0, \tag{25}$$

$$1 - p - 2\lambda u = 0, \tag{26}$$

$$-2\alpha\lambda = 0. \tag{27}$$

Combining (21), (22), (25) we have:

$$\begin{aligned} \dot{x} &= x^2 - \tfrac{3}{2}x + u = f^1, \\ \dot{p} &= p(\tfrac{3}{2} - 2x) = f^2, \end{aligned} \tag{28}$$

with the boundary conditions:

$$x(0) = 1, \quad x(1) = \tfrac{1}{2}.$$

In addition to the boundary-value system given by Eq. (22), (28), the following equations must be satisfied

$$\begin{aligned} 0 &= 1 - u^2 - \alpha^2 = W^3, \\ 0 &= 1 - p - 2\lambda u = W^4, \\ 0 &= \alpha\lambda = W^5. \end{aligned} \tag{29}$$

For the discussion on the application of the Newton operator technique to the system consisting of Eq. (28), (29) we denote that

$$x^1(t) = x(t), \quad x^2(t) = p(t), \quad x^3(t) = u(t), \quad x^4(t) = \alpha(t), \quad x^5(t) = \lambda(t),$$

$$\begin{aligned} f^i &= f^i(x^1, \ldots, x^5, t), \quad i = 1, 2, \\ W^i &= W^i(x^1, \ldots, x^5, t), \quad i = 3, 4, 5. \end{aligned}$$

Setting

$$\begin{aligned} \boldsymbol{x} &= (x^1, \ldots, x^5)^{\mathrm{T}}, \\ \boldsymbol{f} &= (f^1, f^2)^{\mathrm{T}}, \\ \boldsymbol{W} &= (W^3, W^4, W^5)^{\mathrm{T}} \end{aligned}$$

the Eq. (28), (29) may be rewritten as follows

$$\begin{aligned} \dot{\boldsymbol{x}} &= \boldsymbol{f}(\boldsymbol{x}, t), \\ \boldsymbol{0} &= \boldsymbol{W}(\boldsymbol{x}, t). \end{aligned} \tag{30}$$

The algorithm of the original Newton's method requires the solution of the sequence of linear equations:

$$\dot{x}_{n+1} = J(x_n, t)\left[x_{n+1} - x_n\right] + f(x_n, t) \tag{31}$$

with boundary conditions (22) and

$$0 = I(x_n, t)\left[x_{n+1} - x_n\right] + W(x_n, t), \tag{32}$$

$n = 0, 1, \ldots$ where $J(x_n, t)$ is the matrix with elements

$$J_{ij} = \left.\frac{\partial f^i}{\partial x^j}\right|_{x=x_n}, \quad i = 1, 2\,; \quad j = 1, \ldots, 5$$

and

$$I_{ij} = \left.\frac{\partial W^i}{\partial x^j}\right|_{x=x_n}, \quad i = 3, 4, 5\,; \quad j = 1, \ldots, 5$$

Here

$$J(x_n, t) = \begin{pmatrix} 2x_n - \frac{3}{2} & 0 & 1 & 0 & 0 \\ -2p_n & \frac{3}{2} - 2x_n & 0 & 0 & 0 \end{pmatrix}$$

and

$$I(x_n, t) = \begin{pmatrix} 0 & 0 & -2u_n & -2\alpha_n & 0 \\ 0 & -1 & -2\lambda_n & 0 & -2u_n \\ 0 & 0 & 0 & \lambda_n & \alpha_n \end{pmatrix}$$

At every iteration $n + 1$ the function $u_{n+1} = x^3_{n+1} = x^3_{n+1}(x^1_{n+1}, x^2_{n+1})$ is obtained from Eq. (32). This relation is used to eliminate x^3_{n+1} from (31). The function $x^1_{n+1}(t)$, $x^2_{n+1}(t)$ are then computed from the differential equation (31). After them $x^3_{n+1}(t)$, $x^4_{n+1}(t)$, $x^5_{n+1}(t)$ are computed from (32).

The iteration proceeds until $\varrho(x_{n+1}, x_n) \leqq \delta$, where

$$\varrho(x_{n+1}, x_n) = \sum_{i=1}^{2} \max_{t\in[t_0,t_f]} \left|x^i_{n+1} - x^i_n\right| \tag{33}$$

and δ is a suitably small positive constant. The corresponding x_{n+1} is accepted as a solution. The final iteration is then compared with the known closed form solution.

For the problem posed above the starting vector $x(t)$ is chosen as follows:

$$\begin{aligned} x^1_0(t) &= x_0(t) = 1 - t/2\,, \\ x^2_0(t) &= p_0(t) = 1 + 0{\cdot}5t(t-1)\,, \\ x^3_0(t) &= u_0(t) = \cos(3{\cdot}14159t)\,, \\ x^4_0(t) &= \alpha_0(t) = 0{\cdot}3\,, \\ x^5_0(t) &= \lambda_0(t) = 0{\cdot}05\,. \end{aligned}$$

The sequence $\{x_n\}$ converges ($\varrho < 6 \,.\, 10^{-3}$) in 7 iterations. The final value of the state variable $x(t)$ is within 0·005 of the corresponding value for the closed form solution.

The total computer time (IBM 370/135) required is 61 seconds. Fig. 1 and table 2 illustrate the convergence for the control function $u(t)$. The Fig. 1 and the table 1 illustrate the histories of the trajectory $x(t)$.

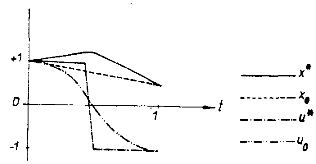

Fig. 1.

We now solve the problem above using the modified Newton's method. The algorithm of the modified Newton's method requires the solution of the sequence of linear equations

$$\dot{x}_{n+1} = J(x_0, t)\left[x_{n+1} - x_n\right] + f(x_n, t)\,, \tag{34}$$

with the boundary conditions (22) and

$$0 = I(x_0, t)\left[x_{n+1} - x_n\right] + W(x_n, t)\,, \quad n = 0, 1, \ldots\,, \tag{35}$$

Table 1.

x \ t	0·0	0·2	0·4	0·6	0·8	1·0
$x_0(t)$	1·0	0·90	0·80	0·70	0·60	0·50
$x_5(t)$	1·000000	1·106212	1·228520	1·117267	0·815438	0·499709
$x_6(t)$	1·000000	1·105902	1·227118	1·110352	0·808306	0·499058
$x_7(t)$	1·000000	1·105788	1·227997	1·113453	0·811805	0·502684
$x_8(t)$	1·000000	1·105903	1·228140	1·116583	0·815106	0·505941

Table 2.

$u(t)$ \ t	0·0	0·2	0·4	0·6	0·8	1·0
$u_5(t)$	1·001946	1·000322	1·001095	−1·000324	−1·004044	−1·094306
$u_6(t)$	1·000001	1·000000	1·000000	−1·000001	−1·000002	−1·006395
$u_7(t)$	0·999992	1·000000	1·000005	−1·000000	−1·000000	−1·000002
$u_8(t)$	1·000007	1·000000	1·000024	−1·000000	−1·000000	−1·000000

where $J(x_0, t)$ is the matrix with elements

$$J_{ij}(x_0, t) = \frac{\partial f^i}{\partial x^j}\bigg|_{x=x_0}, \quad i = 1, 2\,; \quad j = 1, \ldots, 5\,,$$

and

$$I_{ij}(x_0, t) = \frac{\partial W^i}{\partial x^j}\bigg|_{x=x_0}; \quad i = 3, 4, 5\,; \quad j = 1, \ldots, 5\,;$$

here

$$J(x_0, t) = \begin{pmatrix} 2x_0 - \frac{3}{2} & 0 & 1 & 0 & 0 \\ -2p_0 & \frac{3}{2} - 2x_0 & 0 & 0 & 0 \end{pmatrix}$$

and

$$I(x_0, t) = \begin{pmatrix} 0 & 0 & -2u_0 & -2\alpha_0 & 0 \\ 0 & -1 & -2\lambda_0 & 0 & -2u_0 \\ 0 & 0 & 0 & \lambda_0 & \alpha_0 \end{pmatrix}$$

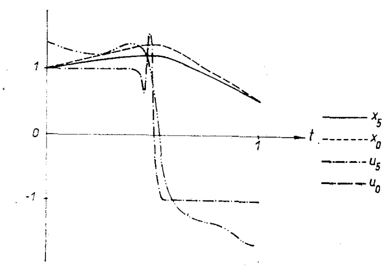

Fig. 2.

and the system equations for using to eliminate x^3_{n+1} from (34) at $(n + 1)$th iteration is

$$0 = -2u_0(u_{n+1} - u_n) - 2\alpha_0(\alpha_{n+1} - \alpha_n) + 1 - u_n^2 - \alpha_n^2\,,$$

$$0 = -p_{n+1} - 2\lambda_0(u_{n+1} - u_n) - 2u_0(\lambda_{n+1} - \lambda_n) + 1 - 2\lambda_n u_n\,,$$

$$0 = \lambda_0\alpha_{n+1} + \alpha_0\lambda_{n+1} - \alpha_n\lambda_n - \lambda_0\alpha_n - \alpha_0\lambda_n\,.$$

In order to carry out the computation the time interval [0, 1] is divided in 50 equal subintervals. The starting values x_0, u_0 and the results are exhibited in the table 3, where for brevity only ten points in time are shown. Fig. 2 illustrates the histories of control function $u(t)$ and the state function $x(t)$. The computer time (IBM 370/135) required is 47 seconds.

Table 3.

x \ t	0·0	0·2	0·4	0·6	0·8	1·0
x_0	1·000000	1·180000	1·350000	1·270000	0·930000	0·500000
x_3	1·000000	1·106742	1·229181	1·121800	0·819904	0·499942
x_4	1·000000	1·106144	1·228434	1·116551	0·814664	0·499964
x_5	1·000000	1·105968	1·228021	1·115644	0·813897	0·499942
u_0	1·420000	1·250000	1·300000	−1·100000	−1·440000	−1·770000
u_3	1·005121	1·000975	1·002108	−1·000216	−1·008883	−1·134105
u_4	1·001508	1·000198	1·000344	−0·999904	−1·002913	−1·083404
u_5	1·000383	0·999980	0·998095	−1·002388	−1·003308	−1·072641

Example 2. Consider the control system described by the Van der Pol's equation

$$\begin{aligned} \dot{x}^1 &= x^2 , \\ \dot{x}^2 &= (1 - (x^1)^2)\, x^2 - x^1 + u , \end{aligned} \tag{36}$$

with the control function $|u(t)| \leqq 1$, and the boundary conditions

$$x^1(0) = 1 , \quad x^2(0) = 0 , \quad x^1(6) = -0{\cdot}85 , \quad x^2(6) = -0{\cdot}93 . \tag{37}$$

It is required to minimize the cost function

$$J(x, u) = \int_0^6 ((x^1)^2 + (x^2)^2 + u^2)\, \mathrm{d}t . \tag{38}$$

We rewritte the inequality constraint on u as

$$1 - u^2 - \alpha^2 = 0 .$$

The notation $\Phi(t)$ will be used for the function

$$\begin{aligned} \Phi(t) = (x^1)^2 + (x^2)^2 + u^2 + p^1(\dot{x}^1 - x^2) + \\ + p^2(\dot{x}^2 + x^1 - (1 - (x^1)^2)\, x^2 - u) + \lambda(1 - u^2 - \alpha^2) . \end{aligned}$$

We now have the Euler equations

$$\begin{aligned} \dot{x}^1 &= x^2 &&= f^1 , \\ \dot{x}^2 &= -x^1 + (1 - (x^1)^2)\, x^2 + u &&= f^2 , \\ \dot{p}^1 &= 2x^1 + 2p^2x^1x^2 + p^2 &&= f^3 , \\ \dot{p}^2 &= -p^1 - p^2(1 - (x^1)^2) + 2x^2 &&= f^4 , \\ 0 &= 1 - u^2 - \alpha^2 &&= W^5 , \\ 0 &= 2u - p^2 - 2\lambda u &&= W^6 , \\ 0 &= \alpha\lambda &&= W^7 . \end{aligned} \tag{39}$$

We denote

$$\begin{aligned} \boldsymbol{x} &= (x^1, x^2, p^1, p^2, u, \alpha, \lambda)\,, \\ \boldsymbol{f} &= (f^1, f^2, f^3, f^4)\,, \\ \boldsymbol{W} &= (W^5, W^6, W^7)\,. \end{aligned}$$

The algorithm of the original Newton's method required the solution of the sequence of linear equations

$$\dot{\boldsymbol{x}}_{n+1} = \boldsymbol{J}(\boldsymbol{x}_n, t)\left[\boldsymbol{x}_{n+1} - \boldsymbol{x}_n\right] + \boldsymbol{f}(\boldsymbol{x}_n, t)\,, \tag{40}$$

with the boundary conditions (37) and

$$\boldsymbol{0} = \boldsymbol{I}(\boldsymbol{x}_n, t)\left[\boldsymbol{x}_{n+1} - \boldsymbol{x}_n\right] + \boldsymbol{W}(\boldsymbol{x}_n, t)\,, \tag{41}$$

$$n = 0, 1, \ldots,$$

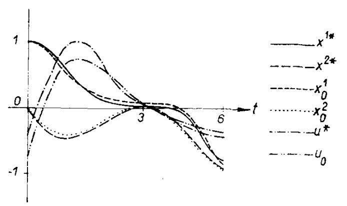

Fig. 3.

Table 4.

t	0·0	1·2	1·4	3·6	4·8	6·0
x_0^1	1·000000	0·510000	0·170000	0·120000	−0·020000	−0·850000
x_3^1	1·000000	0·533218	0·130171	0·077403	−0·039625	−0·849649
x_4^1	1·000000	0·532741	0·131401	0·077881	−0·039645	−0·849562
x_5^1	1·000000	0·531552	0·131934	0·078303	−0·039514	−0·849559
x_0^2	0·0	−0·39	−0·14	0·005	−0·33	−0·93
x_3^2	0·0	−0·486382	−0·160277	0·011878	−0·310252	−0·930258
x_4^2	0·0	−0·480777	−0·160539	0·011222	−0·310404	−0·929842
x_5^2	0·0	−0·479040	−0·160145	0·010918	−0·310553	−0·929953
u_0	−0·73	0·70	0·53	0·08	−0·23	−0·36
u_3	−0·389155	1·033244	0·562008	0·039196	−0·267760	−0·429400
u_4	−0·404046	1·000549	0·562048	0·040638	−0·267189	−0·429323
u_5	−0·410120	1·000001	0·560762	0·041339	−0·266756	−0·429330

 where

$$J(x_n, t) = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ -1 - 2x_n^1 x_n^2 & 1 - (x_n^1)^2 & 0 & 0 & 1 & 0 & 0 \\ 2 + 2p_n^2 x_n^2 & 2p_n^2 x_n^1 & 0 & 2x_n^1 x_n^2 + 1 & 0 & 0 & 0 \\ 2p_n^2 x_n^1 & 2 & -1 & (x_n^1)^2 - 1 & 0 & 0 & 0 \end{pmatrix}$$

and

$$I(x_n, t) = \begin{pmatrix} 0 & 0 & 0 & 0 & -2u_n & -2\alpha_n & 0 \\ 0 & 0 & 0 & -1 & 2 - 2\lambda_n & 0 & -2u_n \\ 0 & 0 & 0 & 0 & 0 & \lambda_n & \alpha_n \end{pmatrix}.$$

To carry out the computation, the time interval $[t_0, t_f]$ is divided in 50 equal subintervals. The starting values x_0^1, x_0^2, u_0 and some values x_n^1, x_n^2, u_n are established in the table 4. Fig. 3 illustrates the histories of the control function $u(t)$ and the state functions $x^1(t)$ and $x^2(t)$.

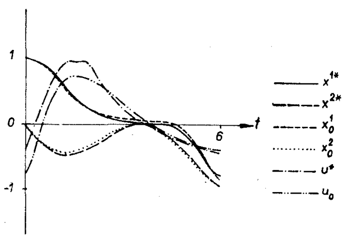

Fig. 4.

Table 5.

t	0·0	1·2	2·4	3·6	4·8	6·0
x_0^1	1·000000	0·510000	0·170000	0·120000	−0·020000	−0·850000
x_{16}^1	1·000000	0·534173	0·130672	0·077397	−0·039570	−0·849581
x_{17}^1	1·000000	0·534382	0·130759	0·077362	−0·039599	−0·849566
x_{18}^1	1·000000	0·534551	0·130839	0·077350	−0·039596	−0·849564
x_0^2	0·0	−0·39	−0·14	0·005	−0·33	−0·93
x_{16}^2	0·0	−0·481383	−0·161073	0·011788	−0·310165	−0·930051
x_{17}^2	0·0	−0·480385	−0·161274	0·011761	−0·310140	−0·930014
x_{18}^2	0·0	−0·479588	−0·161435	0·011748	−0·310118	−0·930049
u_0	−0·73	0·70	0·53	0·08	−0·23	−0·36
u_{16}	−0·395330	1·005159	0·563776	0·039568	−0·267926	−0·429760
u_{17}	−0·396516	1·000141	0·564260	0·039643	−0·267959	−0·429772
u_{18}	−0·397594	0·994736	0·564664	0·039716	−0·267959	−0·429818

The sequence $\{x_n\}$ converges to an accuracy of six significant figures in total 5 iterations.

$$\max |x^{1*} - x_4^1| + \max |x^{2*} - x_4^2| = 0{\cdot}006 .$$

The computer time (IBM 370/135) required is 1 minute. We now solve this problem using the modified Newton's method. The algorithm of the modified Newton's method requires the solution of the sequence of linear equations

$$\dot{x}_{n+1} = J(x_0, t)\,[x_{n+1} - x_n] + f(x_n, t)\,, \tag{42}$$

with the boundary conditions (37) and

$$0 = I(x_0, t)\,[x_{n+1} - x_n] + W(x_n, t)\,, \quad n = 0, 1, \ldots . \tag{43}$$

The time interval [0, 6] is divided in 50 equal subintervals. The results are exhibited in the table 5, where for brevity only six points in time are shown. $x^{1*}(t)$, $x^{2*}(t)$ and $u^*(t)$ are the results obtained from the final iteration. Fig. 4 illustrates the histories of $u(t)$, $x^1(t)$ and $x^2(t)$. The sequence $\{x_n\}$ converges with an accuracy of six significant figures in total 18 iterations. The computer time (IBM 370/135) required is approximately 2 minutes, and

$$\max |x^{1*} - x_{17}^1| + \max |x^{2*} - x_{17}^2| = 0{\cdot}000977 .$$

IV. CONCLUSION

The numerical examples of this paper suggest that for the non-linear optimal control problems with boundary conditions and with constraint on control the modified Newton's method can be effectively used for aceptable solutions.

The main advantages of this method is rapidity of convergence and stable computation. Computer algorithm is simple and fast, storage requirements are small. Therefore these techniques are simple to handle as well as to program.

(Received November 6, 1974.)

REFERENCES

[1] N. Canh: A convergence theorem on the iterative solution of non-linear two-point boundary value systems. Kybernetika *10* (1974), 1, 49–60.

[2] P. Kenneth and G. E. Taylor: Solution of variational problems with bounded control variables by means of the Generalized Newton-Raphson method. Proceedings of the Symposium on Recent advances in optimization technique. J. Wiley, New York 1966, 471–488.

[3] G. M. Lance: Numerical methods for high speed computers. Illife and Sons, London 1960.

Ing. Nguyen Canh, CSc.; Technical University of Hanoi, Hanoi. Democratic Republic of Vietnam. RNDr. Antonín Tuzar, CSc.; Ústav teorie informace a automatizace ČSAV (Institute of Information Theory and Automation — Czechoslovak Academy of Sciences), Pod vodárenskou věží 4, 180 76 Praha 8. Czechoslovakia.