AERO-ASTRONAUTICS REPORT NO. 57

THE RESTORATION OF CONSTRAINTS IN NONHOLONOMIC PROBLEMS: NUMERICAL EXAMPLES

by

JOHN N. DAMOULAKIS

RICE UNIVERSITY

1969

The Restoration of Constraints in Nonholonomic Problems: Numerical Examples[1]

by

JOHN N. DAMOULAKIS[2]

Abstract. In previous papers (Refs. 1 and 2), an iterative procedure was developed in order to restore the differential constraints describing a system. The procedure involves quasilinearization with an added optimality condition, namely, the requirement of least-square change of the control and the state. In this paper, several numerical examples are supplied. They illustrate the rapidly converging characteristics of the algorithm described in Refs. 1 and 2.

[1] This research was supported by the NASA-Manned Spacecraft Center, Grant No. NGR-44-006-089. The author is indebted to Professor Angelo Miele for suggesting the problem and helpful discussions.

[2] Graduate Student in Aero-Astronautics, Department of Mechanical and Aerospace Engineering and Materials Science, Rice University, Houston, Texas.

1. Introduction

In previous papers (Refs. 1-2), a system described by n differential equations of the first order involving n state variables and m control variables was considered. It was assumed that a nominal state x(t) and a nominal control u(t), not satisfying all the equations, but consistent with the boundary conditions, are given. An iterative procedure was developed leading to a varied state $\tilde{x}(t)$ and a varied control $\tilde{u}(t)$ consistent with all the equations and the boundary conditions. In this paper, the procedure developed in Refs. 1 and 2 is applied to solve several restoration problems. The restoration is performed by minimizing the integral

$$J = \frac{1}{2}\int_0^T (\delta u^T \delta u + K\,\delta x^T \delta x)\,dt \tag{1}$$

where $\delta u(t)$ is the control change, $\delta x(t)$ the state change, and K a weighting coefficient. Numerical examples are developed for three values of the weighting coefficient, namely, $K = 0$, $K = 0.5$, and $K = 1$; it is shown that the procedure of Refs. 1 and 2 restores the constraints to the same degree of accuracy in the same number of iterations, regardless of the value of K in the range $0 \le K \le 1$.

2. Theory

Consider a system described by the nonholonomic equation[3]

$$\dot{x} = \varphi(x, u, t) \tag{2}$$

where x (state variable) denotes an n-vector, u (control variable) denotes an m-vector, and φ denotes an n-vector. The independent variable is the time t and the dot denotes the derivative with respect to t. Assume that the state variable is subject to the end conditions

$$x(0) = \alpha, \quad x(\tau) = \beta \tag{3}$$

where τ is prescribed and α and β denote prescribed n-vectors.

Next, suppose that nominal functions x(t), u(t) satisfying the boundary conditions (3), but not consistent with the differential constraint (2) are available. Let $\tilde{x}(t), \tilde{u}(t)$ denote varied functions satisfying the boundary conditions (3) and related to the nominal functions as follows:

$$\tilde{x}(t) = x(t) + \delta x(t) \quad , \quad \tilde{u}(t) = u(t) + \delta u(t) \tag{4}$$

where $\delta x(t)$ and $\delta u(t)$ denote the perturbations of x and u about the nominal values. If quasilinearization is employed, Eq. (2) is approximated by

$$\delta\dot{x} = A^T \delta x + B^T \delta u - (\dot{x} - \varphi) \tag{5}$$

In Eq. (5), A denotes the n x n matrix whose jth column is the gradient of the function φ^j with respect to the vector x; analogously, B denotes the m x n matrix whose jth

---

[3] The vector x has scalar components $x^1, x^2, \ldots, x^n$. The vector u has scalar components $u^1, u^2, \ldots, u^m$. The vector φ has scalar components $\varphi^1, \varphi^2, \ldots, \varphi^n$.

column is the gradient of the function φ^j with respect to the vector u. The superscript T denotes the transpose of a matrix. In order to prevent the perturbations δx and δu from becoming too large, it is convenient to imbed Eq. (5) into the one-parameter family of equations

$$\delta\dot{x} = A^T \delta x + B^T \delta u - \mu(\dot{x} - \varphi) \tag{6}$$

where μ is a prescribed scaling factor in the range

$$0 \leq \mu \leq 1 \tag{7}$$

The boundary conditions (3) become

$$\delta x(0) = 0 \ , \quad \delta x(\tau) = 0 \tag{8}$$

2.1. Minimal Problem. If the functions x(t), u(t) are an approximation to an interesting solution, one may wish to restore the constraint (2) while causing the least-square change of the control and the state. Therefore, we minimize the functional (1), where K is a weighting coefficient, subject to the linearized constraint (6) and the boundary conditions (8). Standard methods of the calculus of variations (Refs. 1-2) show that the Euler equations of this problem are given by[4]

$$\dot{\lambda} = - A\lambda + K\delta x \ , \quad \delta u = B\lambda \tag{9}$$

where $\lambda(t)$, an n-vector, denotes an undetermined, variable Lagrange multiplier. Equations (9) are to be solved in combination with Eq. (6) and the boundary conditions (8). Upon eliminating δu from (6) and (9-2), we obtain the differential system

[4] The vector λ has scalar components $\lambda^1, \lambda^2, \ldots, \lambda^n$.

$$\delta\dot{x} = A^T\delta x + (B^T B)\lambda - \mu(\dot{x} - \varphi) \;, \quad \dot{\lambda} = -A\lambda + K\delta x \tag{10}$$

which, for a given value of μ, must be integrated subject to the boundary conditions (8). Once the functions $\delta x(t)$ and $\lambda(t)$ are known, the function $\delta u(t)$ can be computed from (9-2).

Since Eqs. (10) are linear in δx and λ, any of the methods for solving linear equations with variable coefficients can be employed. For example, let the method of particular solutions be used (Ref. 3). To this effect, we integrate Eqs. (10) forward $n + 1$ times from $t = 0$ to $t = \tau$ using $n + 1$ different sets of initial conditions and the stopping condition $t = \tau$. From these integrations, we obtain the pairs of functions

$$\delta x_i = \delta x_i(t) \;, \quad \lambda_i = \lambda_i(t) \;, \qquad i = 1, 2, \ldots, n+1 \tag{11}$$

each of which is a particular solution of (10). In each integration, the prescribed initial condition (8-1) is employed, that is, $\delta x_i(0)$ is such that

$$\delta x_i(0) = 0 \;, \quad i=1, 2, \ldots, n+1 \tag{12}$$

We note that, for each i, Eq. (12) is equivalent to n scalar conditions. Since 2n initial conditions are needed for each integration of the system (10), Eq. (12) must be completed by the relation

$$\lambda_i(0) = \begin{bmatrix} \delta_{i1} \\ \delta_{i2} \\ \vdots \\ \delta_{in} \end{bmatrix} \;, \quad i=1, 2, \ldots, n+1 \tag{13}$$

where the Kronecker deltas are such that

$$\begin{aligned} \delta_{ij} &= 1 \quad , \quad i = j \\ \delta_{ij} &= 0 \quad , \quad i \neq j \end{aligned} \tag{14}$$

Next, we introduce the n + 1 undetermined, scalar constants k_i and form the linear combinations

$$\delta x(t) = \sum_{i=1}^{n+1} k_i \delta x_i(t) \quad , \quad \lambda(t) = \sum_{i=1}^{n+1} k_i \lambda_i(t) \tag{15}$$

Then, we inquire whether, by an appropriate choice of the constants, these linear combinations can satisfy the differential equations (10) and the end conditions (8). As shown in Ref. 3, this is precisely the case if the constants k_i are determined as follows:

$$\sum_{i=1}^{n+1} k_i = 1 \quad , \quad \sum_{i=1}^{n+1} k_i \delta x_i(\tau) = 0 \tag{16}$$

We note that (16-1) is a scalar equation, while (16-2) is a vector equation, equivalent to n scalar equations. Therefore, the system (16) is equivalent to n + 1 scalar equations which are linear and supply the constants k_i.

2.2. Performance Index: Descent Property. Here, we define the scalar performance index

$$P = \int_0^\tau (\dot{x} - \varphi)^T (\dot{x} - \varphi)\, dt \tag{17}$$

Clearly, P = 0 if x(t) and u(t) satisfy Eq. (2), and P > 0 otherwise. The first variation of the performance index is given by (Ref. 2)

$$\delta P = - 2\mu P \tag{18}$$

Since $P > 0$, Eq. (18) shows that the first variation of the performance index is negative for $\mu > 0$. Therefore, if μ is sufficiently small, the decrease of the performance index is guaranteed.

2.3. Summary of the Algorithm. In the light of the previous discussion, we summarize the algorithm as follows:

(a) Assume nominal functions $x(t)$, $u(t)$.

(b) For the nominal functions, compute the vector $\dot{x} - \varphi$, the matrices A and B, and the performance index P with Eq. (17).

(c) Assuming $\mu = 1$, determine the $n + 1$ particular solutions $\delta x_i(t)$, $\lambda_i(t)$ by forward integration of Eqs. (10) subject to the initial conditions (12)-(13).

(d) Compute the $n + 1$ constants k_i from Eqs. (16).

(e) Determine the correction $\delta x(t)$ with Eq. (15-1), the function $\lambda(t)$ with Eq. (15-2), and the correction $\delta u(t)$ with Eq. (9-2).

(f) Compute the varied functions $\tilde{x}(t)$ and $\tilde{u}(t)$ with Eqs. (4).

(g) For the varied functions $\tilde{x}(t)$ and $\tilde{u}(t)$, compute the performance index $\tilde{P}$. If $\tilde{P} < P$, the scaling factor $\mu = 1$ is acceptable. If $\tilde{P} > P$, the previous value of μ must be replaced by some smaller value in the range (7) until the condition $\tilde{P} < P$ is met. This can be achieved through successive bisections of μ (see Ref. 2).

(h) After a value of μ in the range (7) has been found such that $\tilde{P} < P$, the first iteration is completed. Next, the function $\tilde{x}(t)$, $\tilde{u}(t)$ given by Eqs. (4) are employed as the nominal functions x(t), u(t) for the second iteration, and the procedure is repeated until a desired degree of accuracy is obtained, that is, until the performance index (17) satisfies the inequality

$$P \leq \epsilon \tag{19}$$

where ϵ is a small number.

3. Examples Involving One Differential Constraint

For simplicity, all the symbols employed in this section denote scalar quantities. Consider the scalar differential constraint

$$\dot{x} = \varphi(x, u, t) \tag{20}$$

subject to the boundary conditions

$$x(0) = \alpha \ , \quad x(\tau) = \beta \tag{21}$$

Assume nominal functions $x(t)$, $u(t)$ consistent with the boundary conditions (21) but not consistent with the differential constraint (20). To restore the constraint, the iterative algorithm is represented by

$$\tilde{x}(t) = x(t) + \delta x(t) \quad , \quad \tilde{u}(t) = u(t) + \delta u(t) \tag{22}$$

If $\lambda(t)$ denotes a variable Lagrange multiplier, the functions $\delta x(t)$ and $\lambda(t)$ are obtained by solving the differential system

$$\delta\dot{x} = \varphi_x \delta x + \varphi_u^2 \lambda - \mu(\dot{x} - \varphi) \ , \quad \dot{\lambda} = -\varphi_x \lambda + K\delta x \tag{23}$$

subject to the boundary conditions

$$\delta x(0) = 0 \quad , \quad \delta x(\tau) = 0 \tag{24}$$

For a given μ, the solution of (23) is represented by

$$\delta x(t) = k_1 \delta x_1(t) + k_2 \delta x_2(t) \ , \quad \lambda(t) = k_1 \lambda_1(t) + k_2 \lambda_2(t) \tag{25}$$

The subscripts 1 and 2 denote the particular solutions obtained by forward integration of (23) subject to the initial conditions

$$\begin{aligned} &\underline{\text{First integration:}} \qquad && \delta x_1(0) = 0 \;, \quad \lambda_1(0) = 1 \\ &\underline{\text{Second integration:}} \qquad && \delta x_2(0) = 0 \;, \quad \lambda_2(0) = 0 \end{aligned} \tag{26}$$

The constants k_1 and k_2 are obtained by solving the linear system

$$k_1 + k_2 = 1 \;, \qquad k_1 \delta x_1(\tau) + k_2 \delta x_2(\tau) = 0 \tag{27}$$

After the functions $\delta x(t)$ and $\lambda(t)$ are known, the control change is computed from

$$\delta u(t) = \varphi_u \lambda(t) \tag{28}$$

Example 3.1. Consider the differential constraint

$$\dot{x} = x^2 + u \tag{29}$$

subject to the boundary conditions

$$x(0) = 1 \;, \quad x(1) = 1 \tag{30}$$

Assume the nominal functions

$$x(t) = 1 \;, \quad u(t) = 0 \tag{31}$$

which satisfy the boundary conditions (30) but not the differential constraint (29). To restore the constraint, the algorithm of Section 2 is employed and is repeated until

Ineq. (19) is satisfied for $\varepsilon = 10^{-10}$. Computations performed with a Burroughs B-5500 computer in double-precision arithmetic are summarized in Tables 1-6, where K denotes the weighting coefficient and N denotes the iteration number.

Example 3.2. Consider the differential constraint

$$\dot{x} = x^4 + u \tag{32}$$

subject to the boundary conditions

$$x(0) = 0 \quad , \quad x(1) = 1 \tag{33}$$

Assume the nominal functions

$$x(t) = t \quad , \quad u(t) = 0.5 \tag{34}$$

which satisfy the boundary conditions (33) but not the differential constraint (32). To restore the constraint, the algorithm of Section 2 is employed and is repeated until Ineq. (19) is satisfied for $\varepsilon = 10^{-10}$. Computations performed with a Burroughs B-5500 computer in double-precision arithmetic are summarized in Tables 7-12, where K denotes the weighting coefficient and N denotes the iteration number.

Example 3.3. Consider the differential constraint

$$\dot{x} = x^2 + \exp(u) \tag{35}$$

subject to the boundary conditions

$$x(0) = 0 \quad , \quad x(1) = 3 \tag{36}$$

Assume the nominal functions

$$x(t) = 3t \quad , \quad u(t) = 0 \tag{37}$$

which satisfy the boundary conditions (36) but not the differential constraint (35). To restore the constraint, the algorithm of Section 2 is employed and is repeated until Ineq. (19) is satisfied for $\epsilon = 10^{-10}$. Computations performed with a Burroughs B-5500 computer in double-precision arithmetic are summarized in Tables 13-18, where K denotes the weighting coefficient and N denotes the iteration number.

4. Examples Involving Two Differential Constraints

For simplicity, all the symbols employed in this section denote scalar quantities. Consider the scalar differential constraints

$$\dot{x} = \varphi(x, y, u, t) \quad , \quad \dot{y} = \psi(x, y, u, t) \tag{38}$$

subject to the boundary conditions

$$x(0) = \alpha \quad , \quad y(0) = \beta \quad , \quad x(\tau) = \gamma \quad , \quad y(\tau) = \delta \tag{39}$$

Assume nominal functions x(t), y(t), u(t) consistent with the boundary conditions (39) but not consistent with the differential constraints (38). To restore the constraints, the iterative algorithm is represented by

$$\tilde{x}(t) = x(t) + \delta x(t) \quad , \quad \tilde{y} = y(t) + \delta y(t) \quad , \quad \tilde{u}(t) = u(t) + \delta u(t) \tag{40}$$

If $\lambda(t)$ and $\rho(t)$ denote variable Lagrange multipliers, the functions $\delta x(t)$, $\delta y(t)$, $\lambda(t)$, $\rho(t)$ are obtained by solving the differential system

$$\begin{aligned}
\delta\dot{x} &= \varphi_x \delta x + \varphi_y \delta y + \varphi_u^2 \lambda + \varphi_u \psi_u \rho - \mu(\dot{x} - \varphi) \\
\delta\dot{y} &= \psi_x \delta x + \psi_y \delta y + \varphi_u \psi_u \lambda + \psi_u^2 \rho - \mu(\dot{y} - \psi) \\
\dot{\lambda} &= -\varphi_x \lambda - \psi_x \rho + K\delta x \\
\dot{\rho} &= -\varphi_y \lambda - \psi_y \rho + K\delta y
\end{aligned} \tag{41}$$

subject to the boundary conditions

$$\delta x(0) = 0\,, \quad \delta y(0) = 0\,, \quad \delta x(\tau) = 0\,, \quad \delta y(\tau) = 0 \tag{42}$$

For a given μ, the solution of (41) is represented by

$$\begin{aligned}
\delta x(t) &= k_1 \delta x_1(t) + k_2 \delta x_2(t) + k_3 \delta x_3(t) \\
\delta y(t) &= k_1 \delta y_1(t) + k_2 \delta y_2(t) + k_3 \delta y_3(t) \\
\lambda(t) &= k_1 \lambda_1(t) + k_2 \lambda_2(t) + k_3 \lambda_3(t) \\
\rho(t) &= k_1 \rho_1(t) + k_2 \rho_2(t) + k_3 \rho_3(t)
\end{aligned} \tag{43}$$

The subscripts 1, 2, 3 denote the particular solutions obtained by forward integration of (41) subject to the initial conditions

$$\begin{array}{ll}
\underline{\text{First integration:}} & \delta x_1(0) = 0\,, \quad \delta y_1(0) = 0\,, \quad \lambda_1(0) = 1\,, \quad \rho_1(0) = 0 \\
\underline{\text{Second integration:}} & \delta x_2(0) = 0\,, \quad \delta y_2(0) = 0\,, \quad \lambda_2(0) = 0\,, \quad \rho_2(0) = 1 \\
\underline{\text{Third integration:}} & \delta x_3(0) = 0\,, \quad \delta y_3(0) = 0\,, \quad \lambda_3(0) = 0\,, \quad \rho_3(0) = 0
\end{array} \tag{44}$$

After the functions $\delta x(t)$, $\delta y(t)$, $\lambda(t)$, $\rho(t)$ are known, the control change is computed from

$$\delta u(t) = \varphi_u \lambda(t) + \psi_u \rho(t) \tag{45}$$

<u>Example 4.1.</u> Consider the differential constraints

$$\dot{x} = y\,, \quad \dot{y} = 2 \sin u - 1 \tag{46}$$

subject to the boundary conditions

$$x(0) = 0 \,, \quad y(0) = 0 \,, \quad x(1) = 0.3 \,, \quad y(1) = 0 \tag{47}$$

Assume the nominal functions

$$x(t) = 0.3t \,, \quad y(t) = 0 \,, \quad u(t) = 0 \tag{48}$$

which satisfy the boundary conditions (47) but not the differential constraints (46). To restore the constraints, the algorithm of Section 2 is employed and is repeated until Ineq. (19) is satisfied for $\epsilon = 10^{-10}$. Computations performed with a Burroughs B-5500 computer in double-precision arithmetic are summarized in Tables 19-24, where K denotes the weighting coefficient and N denotes the iteration number.

Example 4.2. Consider the differential constraints

$$\dot{x} = y + u \,, \quad \dot{y} = x^2 + \exp(y) \tag{49}$$

subject to the boundary conditions

$$x(0) = 0 \,, \quad y(0) = 0 \,, \quad x(0.5) = 1 \,, \quad y(0.5) = 3 \tag{50}$$

Assume the nominal functions

$$x(t) = 2t \,, \quad y(t) = 6t \,, \quad u(t) = 0 \tag{51}$$

which satisfy the boundary conditions (50) but not the differential constraints (49). To restore the constraints, the algorithm of Section 2 is employed and is repeated until

Ineq. (19) is satisfied for $\epsilon = 10^{-10}$. Computations performed with a Burroughs B-5500 computer in double-precision arithmetic are summarized in Tables 25-30 where K denotes the weighting coefficient and N the iteration number.

Table 1. Converged values of the functions x(t), u(t) for K = 0

(Example 3.1)

t	x(t)	u(t)
0.0	1.0000	-3.5233
0.1	0.7774	-2.5263
0.2	0.6147	-1.7152
0.3	0.5085	-1.0536
0.4	0.4539	-0.5128
0.5	0.4454	-0.0698
0.6	0.4782	0.2939
0.7	0.5492	0.5937
0.8	0.6574	0.8415
0.9	0.8054	1.0472
1.0	1.0000	1.2185

Table 2. The functions P(N), J(N) for K = 0

(Example 3.1)

N	P	μ	J
0	0.90×10^{1}		
1	0.32×10^{-1}	1.0	0.34×10^{1}
2	0.79×10^{-6}	1.0	0.97×10^{-2}
3	0.98×10^{-16}	1.0	0.21×10^{-6}

Table 3. Converged values of the functions x(t), u(t) for K = 0.5 (Example 3.1)

t	x(t)	u(t)
0.0	1.0000	-3.4413
0.1	0.7857	-2.4639
0.2	0.6297	-1.6766
0.3	0.5281	-1.0401
0.4	0.4756	-0.5233
0.5	0.4669	-0.1014
0.6	0.4977	0.2455
0.7	0.5650	0.5335
0.8	0.6686	0.7754
0.9	0.8111	0.9817
1.0	1.0000	1.1607

Table 4. The functions P(N), J(N) for K = 0.5 (Example 3.1)

N	P	μ	J
0	0.90×10^{1}		
1	0.28×10^{-1}	1.0	0.34×10^{1}
2	0.53×10^{-6}	1.0	0.85×10^{-2}
3	0.39×10^{-16}	1.0	0.15×10^{-6}

Table 5. Converged values of the functions x(t), u(t) for K = 1 (Example 3.1)

t	x(t)	u(t)
0.0	1.0000	-3.3649
0.1	0.7931	-2.4058
0.2	0.6433	-1.6407
0.3	0.5460	-1.0275
0.4	0.4956	-0.5330
0.5	0.4869	-0.1306
0.6	0.5157	0.2006
0.7	0.5798	0.4777
0.8	0.6790	0.7139
0.9	0.8165	0.9205
1.0	1.0000	1.1066

Table 6. The functions P(N), J(N) for K = 1 (Example 3.1)

N	P	μ	J
0	0.90×10^{1}		
1	0.24×10^{-1}	1.0	0.34×10^{1}
2	0.37×10^{-6}	1.0	0.74×10^{-2}
3	0.16×10^{-16}	1.0	0.10×10^{-6}

Table 7. Converged values of the functions x(t), u(t) for K = 0 (Example 3.2)

t	x(t)	u(t)
0.0	0.0000	0.9008
0.1	0.0900	0.9007
0.2	0.1801	0.9001
0.3	0.2703	0.8975
0.4	0.3608	0.8906
0.5	0.4520	0.8764
0.6	0.5448	0.8519
0.7	0.6408	0.8150
0.8	0.7430	0.7658
0.9	0.8579	0.7076
1.0	1.0000	0.6472

Table 8. The functions P(N), J(N) for K = 0 (Example 3.2)

N	P	μ	J
0	0.16×10^{0}		
1	0.32×10^{-4}	1.0	0.61×10^{-1}
2	0.79×10^{-11}	1.0	0.61×10^{-5}

Table 9. Converged values of the functions x(t), u(t) for K = 0.5 (Example 3.2)

t	x(t)	u(t)
0.0	0.0000	0.9064
0.1	0.0906	0.9061
0.2	0.1812	0.9048
0.3	0.2718	0.9010
0.4	0.3626	0.8924
0.5	0.4539	0.8761
0.6	0.5467	0.8492
0.7	0.6424	0.8100
0.8	0.7442	0.7589
0.9	0.8585	0.7001
1.0	1.0000	0.6408

Table 10. The functions P(N), J(N) for K = 0.5 (Example 3.2)

N	P	μ	J
0	0.16×10^{0}		
1	0.30×10^{-4}	1.0	0.61×10^{-1}
2	0.60×10^{-11}	1.0	0.57×10^{-5}

Table 11. Converged values of the functions x(t), u(t) for K = 1 (Example 3.2)

t	x(t)	u(t)
0.0	0.0000	0.9116
0.1	0.0911	0.9111
0.2	0.1822	0.9091
0.3	0.2732	0.9042
0.4	0.3642	0.8941
0.5	0.4557	0.8759
0.6	0.5484	0.8468
0.7	0.6439	0.8053
0.8	0.7453	0.7525
0.9	0.8591	0.6929
1.0	1.0000	0.6348

Table 12. The functions P(N), J(N) for K = 1 (Example 3.2)

N	P	μ	J
0	0.16×10^{0}		
1	0.27×10^{-4}	1.0	0.62×10^{-1}
2	0.46×10^{-11}	1.0	0.54×10^{-5}

Table 13. Converged values of the functions $x(t)$, $u(t)$ for $K = 0$ (Example 3.3)

t	x(t)	u(t)
0.0	0.0000	0.5010
0.1	0.1654	0.4913
0.2	0.3331	0.4613
0.3	0.5058	0.4104
0.4	0.6875	0.3421
0.5	0.8849	0.2650
0.6	1.1097	0.1899
0.7	1.3812	0.1255
0.8	1.7324	0.0766
0.9	2.2260	0.0437
1.0	3.0000	0.0243

Table 14. The functions $P(N)$, $J(N)$ for $K = 0$ (Example 3.3)

N	P	u	J
0	0.82×10^{1}		
1	0.17×10^{0}	1.0	0.14×10^{0}
2	0.35×10^{-2}	1.0	0.18×10^{-1}
3	0.16×10^{-5}	1.0	0.56×10^{-3}
4	0.43×10^{-12}	1.0	0.30×10^{-6}

Table 15. Converged values of the functions x(t), u(t) for K = 0.5 (Example 3.3)

t	x(t)	u(t)
0.0	0.0000	0.5390
0.1	0.1716	0.5257
0.2	0.3447	0.4851
0.3	0.5211	0.4181
0.4	0.7045	0.3306
0.5	0.9017	0.2338
0.6	1.1248	0.1413
0.7	1.3934	0.0645
0.8	1.7410	0.0102
0.9	2.2305	-0.0186
1.0	3.0000	-0.0207

Table 16. The functions P(N), J(N) for K = 0.5 (Example 3.3)

N	P	u	J
0	0.82×10^{1}		
1	0.17×10^{0}	1.0	0.19×10^{0}
2	0.35×10^{-2}	1.0	0.19×10^{-1}
3	0.17×10^{-5}	1.0	0.54×10^{-3}
4	0.46×10^{-12}	1.0	0.29×10^{-6}

Table 17. Converged values of the functions x(t), u(t) for K = 1 (Example 3.3)

t	x(t)	u(t)
0.0	0.0000	0.5722
0.1	0.1772	0.5556
0.2	0.3551	0.5057
0.3	0.5348	0.4244
0.4	0.7197	0.3194
0.5	0.9168	0.2044
0.6	1.1383	0.0957
0.7	1.4044	0.0068
0.8	1.7488	-0.0530
0.9	2.2345	-0.0786
1.0	3.0000	-0.0642

Table 18. The functions P(N), J(N) for K = 1 (Example 3.3)

N	P	u	J
0	0.82×10^{1}		
1	0.18×10^{0}	1.0	0.24×10^{0}
2	0.35×10^{-2}	1.0	0.19×10^{-1}
3	0.18×10^{-5}	1.0	0.53×10^{-3}
4	0.54×10^{-12}	1.0	0.30×10^{-6}

Table 19. Converged values of the functions x(t), y(t), u(t) for K = 0 (Example 4.1)

t	x(t)	y(t)	u(t)
0.0	0.0000	0.0000	1.5602
0.1	0.0049	0.0999	1.5193
0.2	0.0199	0.1990	1.4384
0.3	0.0447	0.2952	1.3087
0.4	0.0787	0.3828	1.1232
0.5	0.1206	0.4513	0.8788
0.6	0.1678	0.4847	0.5795
0.7	0.2158	0.4643	0.2387
0.8	0.2584	0.3761	-0.1207
0.9	0.2886	0.2178	-0.4711
1.0	0.3000	0.0000	-0.7871

Table 20. The functions P(N), J(N) for K = 0 (Example 4.1)

N	P	μ	J
0	0.10×10^{1}		
1	0.79×10^{-1}	1.0	0.26×10^{0}
2	0.32×10^{-2}	1.0	0.29×10^{-1}
3	0.31×10^{-4}	1.0	0.28×10^{-2}
4	0.17×10^{-8}	1.0	0.21×10^{-4}
5	0.38×10^{-17}	1.0	0.97×10^{-9}

Table 21. Converged values of the functions x(t), y(t), u(t) for K = 0.5 (Example 4.1)

t	x(t)	y(t)	u(t)
0.0	0.0000	0.0000	1.5647
0.1	0.0049	0.0999	1.5256
0.2	0.0199	0.1991	1.4420
0.3	0.0447	0.2953	1.3071
0.4	0.0787	0.3826	1.1173
0.5	0.1206	0.4505	0.8727
0.6	0.1676	0.4833	0.5786
0.7	0.2155	0.4635	0.2451
0.8	0.2581	0.3770	-0.1119
0.9	0.2885	0.2196	-0.4722
1.0	0.3000	0.0000	-0.8113

Table 22. The functions P(N), J(N) for K = 0.5 (Example 4.1)

N	P	μ	J
0	0.10×10^{1}		
1	0.82×10^{-1}	1.0	0.28×10^{0}
2	0.34×10^{-2}	1.0	0.30×10^{-1}
3	0.35×10^{-4}	1.0	0.30×10^{-2}
4	0.21×10^{-8}	1.0	0.23×10^{-4}
5	0.57×10^{-17}	1.0	0.11×10^{-8}

Table 23. Converged values of the functions x(t), y(t), u(t) for K = 1 (Example 4.1)

t	x(t)	y(t)	u(t)
0.0	0.0000	0.0000	1.5679
0.1	0.0049	0.0999	1.5317
0.2	0.0199	0.1992	1.4457
0.3	0.0447	0.2955	1.3057
0.4	0.0787	0.3825	1.1116
0.5	0.1206	0.4497	0.8669
0.6	0.1675	0.4819	0.5777
0.7	0.2152	0.4627	0.2514
0.8	0.2579	0.3778	-0.1033
0.9	0.2884	0.2213	-0.4733
1.0	0.3000	0.0000	-0.8350

Table 24. The functions P(N), J(N) for K = 1 (Example 4.1)

N	P	μ	J
0	0.10×10^{1}		
1	0.84×10^{-1}	1.0	0.31×10^{0}
2	0.36×10^{-2}	1.0	0.31×10^{-1}
3	0.39×10^{-4}	1.0	0.31×10^{-2}
4	0.26×10^{-8}	1.0	0.25×10^{-4}
5	0.86×10^{-17}	1.0	0.14×10^{-8}

Table 25. Converged values of the functions x(t), y(t), u(t) for K = 0 (Example 4.2)

t	x(t)	y(t)	u(t)
0.0	0.0000	0.0000	10.7408
0.1	1.0192	0.1425	8.9413
0.2	1.7231	0.4795	4.5846
0.3	1.9119	1.0405	-1.8971
0.4	1.5955	1.7724	-6.8427
0.5	1.0000	3.0000	-9.2385

Table 26. The functions P(N), J(N) for K = 0 (Example 4.2)

N	P	u	J
0	0.15×10^{2}		
1	0.85×10^{1}	0.5	0.25×10^{2}
2	0.84×10^{-1}	1.0	0.15×10^{1}
3	0.11×10^{-3}	1.0	0.12×10^{0}
4	0.29×10^{-9}	1.0	0.19×10^{-3}
5	0.20×10^{-20}	1.0	0.51×10^{-9}

Table 27. Converged values of the functions x(t), y(t), u(t) for K = 0.5 (Example 4.2)

t	x(t)	y(t)	u(t)
0.0	0.0000	0.0000	10.7454
0.1	1.0195	0.1425	8.9422
0.2	1.7233	0.4796	4.1616
0.3	1.9116	1.0406	-1.9006
0.4	1.5952	1.7724	-6.8405
0.5	1.0000	3.0000	-9.2369

Table 28. The functions P(N), J(N) for K = 0.5 (Example 4.2)

N	P	μ	J
0	0.15×10^{2}		
1	0.85×10^{1}	0.5	0.25×10^{2}
2	0.84×10^{-1}	1.0	0.16×10^{1}
3	0.11×10^{-3}	1.0	0.12×10^{0}
4	0.29×10^{-9}	1.0	0.19×10^{-3}
5	0.20×10^{-20}	1.0	0.52×10^{-9}

Table 29. Converged values of the functions x(t), y(t), u(t) for K = 1 (Example 4.2)

t	x(t)	y(t)	u(t)
0.0	0.0000	0.0000	10.7500
0.1	1.0199	0.1426	8.9432
0.2	1.7234	0.4797	4.1578
0.3	1.9114	1.0407	-1.9040
0.4	1.5949	1.7725	-6.8384
0.5	1.0000	3.0000	-9.2353

Table 30. The functions P(N), J(N) for K = 1 (Example 4.2)

N	P	u	J
0	0.15×10^{2}		
1	0.85×10^{1}	0.5	0.26×10^{2}
2	0.84×10^{-1}	1.0	0.16×10^{1}
3	0.11×10^{-3}	1.0	0.12×10^{0}
4	0.29×10^{-9}	1.0	0.19×10^{-3}
5	0.20×10^{-20}	1.0	0.53×10^{-9}

References

1. MIELE, A., and DAMOULAKIS, J.N., The Restoration of Constraints in Nonholonomic Problems, Rice University, Aero-Astronautics Report No. 40, 1968.

2. MIELE, A., HEIDEMAN, J.C., and DAMOULAKIS, J.N., The Restoration of Constraints in Holonomic and Nonholonomic Problems, Journal of Optimization Theory and Applications, Vol. 3, No. 5, 1969.

3. MIELE, A., Method of Particular Solutions for Linear, Two-Point Boundary-Value Problems, Journal of Optimization Theory and Applications, Vol. 2, No. 4, 1968.