'93 한국자동제어학술회의논문집 (1993. 10. 20~22)

제한조건을 가진 로봇 매니퓰레이터에 대한 최적 시간 운동

°정일권, 이주장
한국과학기술원 전기 및 전자공학과

Time-Optimal Motions of Robotic Manipulators with Constraints

°Il-Kwon Jeong, Ju-Jang Lee
Department of Electrical Engineering, KAIST

Abstract

In this paper, methods for computing the time-optimal motion of a robotic manipulator are presented that considers the nonlinear manipulator dynamics, actuator constraints, joint limits, and obstacles. The optimization problem can be reduced to a search for the time-optimal path in the n-dimensional position space. These paths are further optimized with a local path optimization to yield a global optimal solution. Path constrained time-optimal motion is also described. Time-optimal motion of a robot with an articulated arm is presented as an example.

1. Introduction

The productivity of robotic systems can be maximized by planning robot motions to be time optimal. Optimal motions can yield reduced cycle times, increased system utilization, and thus improve the cost effectiveness of typical automated manufacturing systems.

Using the optimality condition stated by the Pontryagin maximum principle, the optimization problem can be transformed to a $4n$-dimensional two-point boundary value problem (TPBVP) for an n-degree-of-freedom (DOF) manipulator.

A dynamic programming search over the entire $2n$-dimensional state space can potentially yield the global optimal trajectory among obstacles.

One approach to reduce the complexity of the problem and to facilitate a practical realization of time-optimal motion planning is to represent robot motions by a *path* and a *velocity profile* along the path. This separation allows reducing the optimization problem to two smaller problems: 1) computing the optimal velocity profile along a given path, and 2) searching for the optimal path in the n-dimensional position space. The local optimal path can be obtained with a parameter optimization that iterates on path parameters, such as the control points of a B spline [4].

In this paper, two time-optimal motions are presented. Global search and solving TPBVP with an example of 2-link planar arm robot. Global search requires local optimization technique, path constrained time-optimal motion. In global search, the optimization method is reduced to selecting the best path out of a large (but finite) number of paths represented by a tessellated position space, using a branch and bound search and a series of lower bound estimates on the traveling time along each path.

The local optimization uses a penalty function to ensure that the optimal path does not approach obstacles or manipulator joint limits, and this procedure requires computing for path-constrained time-optimal motions.

An example of 2-link manipulator is presented and solved. The validity of global search is discussed. The solution represented in this paper is global time-optimal since the TPBVP is solved directly.

2. Global Search

2.1 *Work Space Representation*

The workspace is represented by a uniform grid to reduce the number of feasible paths between the end points to a final set, and to facilitate a combinatorial search for the set of best paths. Eliminating paths with loops and sharp turns contributes to the computational efficiency of this approach by drastically reducing the number of possible paths along the grid.

A typical grid point x_i is connected to its adjacent neighbors x_i^j, defined as

$$x_i^j = x_i + \Lambda_j d, \qquad j = 1,\ldots,3^n - 1 \qquad (2.1)$$

where $d \in R^n$ is a vector of the typical grid sizes along the n axes, and $\Lambda_j \in R^{n\times n}$ is a diagonal matrix with the elements -1,

0, or 1.

Augmenting the position space with a direction state allows representing the search by a directed graph with nodes $\{a_{i,j}\}$ and edges $\{e_{i,j}\}$. A typical node $\{a_{i,j} \in x_i, j = 1,\ldots,3^{n-1}\}$ represents grid point x_i and the jth direction of arrival to that point. A typical edge $e_{i,k}$ connects a node belonging to grid point x_i to its neighbor in the k direction.

The complexity of a graph search is proportional to the number of edges. Here, the total number of nodes is $(3^n - 1)m^n$, where n is the number of states, and m is the number of points defined along each state (assuming the same tessellation for all states). The total number of edges is, therefore, $q(3^n - 1)m^n$, where q is the number of departing edges from each node. A state space grid for the same problem would have m^{2n} nodes and $(3^n - 1)m^{2n}$ edges. Dividing the number of nodes of the augmented position space grid yields an exponential improvement of m^n / q in the complexity of the search algorithm.

2.2 *Obstacle Representation*

To allow a search for paths in the Cartesian space, we define *obstacle shadows* as regions formed by grid points that are not accessible to the manipulator tip due to the presence of obstacles. Obstacle shadows can be represented as a mapping of the *configuration space obstacles* to the Cartesian space. Define A as the set of all reachable points in the Cartesian space, mapped from the joint space B by the single valued forward kinematics function $FK(\cdot)$:

$$A = \{x | x = FK(y), y \in B\} \tag{2.2}$$

It is convenient to subdivide the joint space B to subsets B_i consisting each of points for which a single valued inverse kinematic solution exists. For two different points $y, z \in B$ and $y \in B_i$

$$FK(y) = FK(z) \text{ only if } z \notin B_i \tag{2.3}$$

If B_i maps to A_i through (2.2), then the kinematic map between B_i and A_i is invertible.

A *configuration space obstacle* $CO(b)$ due to obstacle b is defined as [1]:

$$CO(b) = \{y | R(y) \cap b \neq 0\} \tag{2.4}$$

where $R(y)$ represents the set of points occupied by robot links at joint positions $y \in B$. We define the ith obstacle shadow $OS(b)_i \in A$ due to obstacle b as

$$OS(b)_i = \{x | x = FK(y), y \in CO(b) \cap B_i\} \tag{2.5}$$

2.3 *Branch and Bound Search*

We use a branch and bound search to select a set of best paths from all possible ones between the end points. The lower bounds are used to branch the search toward the most promising subsets and to discard certain subsets from further consideration. The search is terminated when each subset has been shown to contain no better solution than the one already at hand. The best solution found during this search is a global optimum.

Key to this approach is the selection of appropriate approximations of the cost function that are guaranteed to produce lower bounds on the traveling time along a given path or a set of paths. Several such approximations (called tests) are presented that are based on the physical and dynamic characteristics of the manipulator and its time-optimal paths. The most conservative but efficient approximations are used first, when the number of path candidates is large, and the more accurate but computationally expensive are used last.

2.4 *Lower Bound Tests*

We consider four tests for lower bound estimates on the motion time along a given path. The tests are structures to produce successively higher lower bounds so that the velocity profile of every successive test is tangent and below the previous velocity profile. Hence

$$t_1 \le t_2 \le t_3 \le t_4 = t_{opt}$$

where t_1 is the most conservative but computationally efficient estimate, and t_{opt} is the optimal traveling time along the path.

1) Maximum Speed Test :

The traveling time along a typical path is obtained by the summation

$$t_1 = \sum_j \frac{\Delta x_j}{V_{\max}} \tag{2.6}$$

where Δx_j is the length of the jth grid segment along the path, and $V_{\max}$ is the assigned maximum speed selected as the highest speed along a velocity limit curve for a representative path.

2) Velocity Limit Test :

We assume that the speed along the path follows the velocity limit curve. The lower bound t_2 is obtained by the integral

$$t_2 = \int_{\lambda_0}^{\lambda_f} \frac{1}{\dot{\lambda}_m} d\lambda \tag{2.7}$$

where $\dot{\lambda}_m(\lambda)$ is the velocity limit curve.

3) Maximum Acceleration, Velocity Limit, Maximum Deceleration Test :

This test considers the actual speeds at the end points, which without loss of generality are chosen to be zero.

$$t_3 = \int_{\lambda_0}^{\lambda_1} \frac{d\lambda}{\dot{\lambda}_a(\lambda)} + \int_{\lambda_1}^{\lambda_2} \frac{d\lambda}{\dot{\lambda}_m(\lambda)} + \int_{\lambda_2}^{\lambda_f} \frac{d\lambda}{\dot{\lambda}_d(\lambda)} \tag{2.8}$$

where $\dot{\lambda}_a$ and $\dot{\lambda}_d$ are the speeds along the path during maximum acceleration and maximum deceleration, respectively. λ_1 and λ_2 are the points along the path where $\dot{\lambda}_a$ and $\dot{\lambda}_a$ reach the velocity limit $\dot{\lambda}_m$.

4) Optimal Velocity Along the Path :

This test computes the time-optimal velocity profile along the path. The optimal velocity profile is always below the limit curve and in most cases tangents to the limit curve at a finite number of points.

The tests outlined above are found most useful for articulated manipulators in optimizing motion time.

2.5 *Upper Bound*

The upper bound t_u is used to discard costly subsets early in the branch and bound search. The upper bound t_u is selected below the optimal motion time along the best grid path at hand $\tilde{t}$: $t_u \le \tilde{t} + \varepsilon$, where ε is a constant determined by the shape of the cost function near the global minimum and the grid size. This constant can be determined by evaluating the sensitivity of the cost function to path parameters.

2.6 *Local Optimization*

The optimization problem is formulated as an unconstrained parameter optimization, using the control points of cubic B splines as the optimization variables and the motion time along a specified path as the cost function. Obstacles are represented by penalty functions based on the distance between the manipulator links and the obstacles. To reduce the computation time and improve the convergence of the local optimization, the number of control points is reduced. The true optimum can be approached by successively increasing the number of control points and repeating the local optimization.

Assuming that all paths in a close neighborhood converge to the same optimum, only the best path in each region can be selected as a candidate for the local optimization. A region is defined as a tube of some radius D_{max} around a given path. Starting with the best path obtained by the branch and bound search, all paths in a tube around the best one are discarded. The selection of D_{max} is based on the anticipated size of the convergence region around the global optimum, obtained by the sensitivity test described earlier.

3. Path-Constrained Time-Optimal Motions

3.1 *Parameterized Robot Dynamics with Input Torque Constraints*

Usually the dynamic equations take the form

$$u_i = J_{ij}(q)\ddot{q}^j + R_{ij}\dot{q}^j + C_{ijk}(q)\dot{q}^j\dot{q}^k + G_i(q) \tag{3.1}$$

where u_i is the ith generalized force, q^i is the ith generalized coordinate, J_{ij} the inertia matrix, G_i the gravitational force on the ith joint, C_{ijk} the coriolis force array, and R_{ij} is the viscous friction matrix. The Einstein summation convention has been used, and all indexes run from one to n inclusive for an n-degree-of-freedom robot.

It will be assumed that the path is given as a *parameterized curve*. The curve is assumed to be given by a set of n functions of a single parameter λ, so that we are given

$$q^i = f^i(\lambda), \qquad 0 \le \lambda \le \lambda_{max} \tag{3.2}$$

where λ is a parameter for describing the desired path, and it is assumed that the coordinates q^i vary continuously with λ and that the path never retraces itself as λ goes from 0 to λ_{max}. We differentiate q^i with respect to time,

$$\dot{q}^i = \frac{df^i}{d\lambda}\frac{d\lambda}{dt} = \frac{df^i}{d\lambda}\dot{\lambda} = \frac{df^i}{d\lambda}\mu \tag{3.3}$$

where $\mu \equiv \dot{\lambda}$. The equations of motion along the curve (i.e., the geometric path) then become

$$\dot{\lambda} = \mu \tag{3.4a}$$

$$u_i = J_{ij}(\lambda)\frac{df^j}{d\lambda}\mu + J_{ij}(\lambda)\frac{d^2f^j}{d\lambda^2}\mu^2 + G_i(\lambda) + R_{ij}\frac{df^j}{d\lambda}\mu + C_{ijk}(\lambda)\frac{df^j}{d\lambda}\frac{df^k}{d\lambda}\mu^2 \tag{3.4b}$$

Note that if λ is used to represent arc length along the path, then μ and $\dot{\mu}$ are the velocity and the acceleration along the path, respectively.

The state equations become

$$\dot{\lambda} = \mu \tag{3.5a}$$

$$\dot{\mu} = \frac{1}{J_{ij}(\lambda)\dfrac{df^i}{d\lambda}\dfrac{df^j}{d\lambda}}\left[u_i\frac{df^i}{d\lambda} - J_{ij}(\lambda)\frac{df^i}{d\lambda}\frac{d^2f^j}{d\lambda^2}\mu^2 - G_i(\lambda)\frac{df^i}{d\lambda} - R_{ij}\frac{df^i}{d\lambda}\frac{df^j}{d\lambda}\mu - C_{ijk}(\lambda)\frac{df^i}{d\lambda}\frac{df^j}{d\lambda}\frac{df^k}{d\lambda}\mu^2\right] \tag{3.5b}$$

For evaluating the bounds on $\dot{\mu}$ explicitly, (3.4b) can be plugged in to the inequalities

$u^i_{min} \le u_i \le u^i_{max}$ so that

$$u^i_{min} \le J_{ij}\frac{df^j}{d\lambda}\dot{\mu} + \left(J_{ij}\frac{d^2f^j}{d\lambda^2} + C_{ijk}\frac{df^j}{d\lambda}\frac{df^k}{d\lambda}\right)\mu^2 + R_{ij}\frac{df^j}{d\lambda}\mu + G_i \le u^i_{max} \tag{3.6}$$

3.2 *Optimal Trajectory*

Given the form for the dynamic equations, we have the minimum time path planning problem as follows.

Problem : Find $x^* = (\lambda^*, \mu^*)$ and u_i^* by minimizing T subject to (3.5a), (3.5b),

$u_{i\min}^l \le u_i \le u_{i\max}^l$, $0 \le \lambda \le \lambda_{max}$, and the boundary conditions $\mu(0) = \mu_0$, $\mu(t_f) = \mu_f$, $\lambda(0) = 0$, and $\lambda(t_f) = \lambda_{max}$.

By defining the functions M, Q, R, S, U, the state equations can be rewritten in the following form.

$$\dot{\lambda} = \mu \tag{3.7a}$$

$$\dot{\mu} = \frac{1}{M}[U - Q\mu^2 - R\mu - S] \tag{3.7b}$$

It is instructive to look at the system's behavior in the phase plane. The equations of the phase-plane trajectories can be obtained by dividing (3.7b) by (3.7a). This gives

$$\frac{d\mu}{d\lambda} = \frac{\frac{d\mu}{dt}}{\frac{d\lambda}{dt}} = \frac{\dot{\mu}}{\mu} = \frac{1}{\mu M}[U - Q\mu^2 - R\mu - S] \tag{3.8}$$

The total time T it takes to go from initial to final states is

$$T = \int_o^{t_f} dt = \int_0^{\lambda_{max}} \frac{dt}{d\lambda} d\lambda = \int_0^{\lambda_{max}} \frac{1}{\mu} d\lambda \tag{3.9}$$

The constraints on $\dot{\mu}$ have two effects. One effect is to place limits on the slope of the phase trajectory. The other is to place limits on the value of μ. To obtain the limits on $d\mu / d\lambda$, we simply divide the limits on $\dot{\mu}$ by μ, since $d\mu / d\lambda = \dot{\mu} / \mu$.

With the velocity limit curve and the phase portrait showing acceleration and deceleration vectors at each state, we can determine optimal trajectories by a certain algorithm [6].

4. Time-Optimal Motion of a Robot with an Articulated Arm

4.1 *The Model*

The model is a planar 2-link manipulator. Since its configuration is simple sketch of it is omitted. Joint 1 angle is φ and joint 2 angle is ψ. Torques are M_φ, M_ψ respectively.

With the state variables

$$x_1(t) = \varphi(t) \tag{4.1}$$

$$x_2(t) = \dot{\varphi}(t) \tag{4.2}$$

$$x_3(t) = \psi(t) \tag{4.3}$$

$$x_4(t) = \dot{\psi}(t) \tag{4.4}$$

and the control variables

$$u_1(t) = M_\varphi(t) \tag{4.5}$$

$$u_2(t) = M_\psi(t) \tag{4.6}$$

and neglecting friction the equation of motion are

$$\dot{x}_1 = x_2 \tag{4.7}$$

$$\dot{x}_2 = \frac{J_7\{u_1 - u_2 + J_6(x_2 + x_4)^2 \sin(x_3)\}}{J_7 J_5 - J_6^2 \cos^2(x_3)}$$

$$\underline{\quad - J_6\{u_2 - J_6 x_2^2 \sin(x_3)\}\cos(x_3) \quad} \tag{4.8}$$

$$\dot{x}_3 = x_4 \tag{4.9}$$

$$\dot{x}_4 = \frac{(J_5 + J_6\cos(x_3))\{u_2 - J_6 x_2^2 \sin(x_3)\}}{J_7 J_5 - J_6^2 \cos^2(x_3)}$$

$$\underline{\quad - (J_7 + J_6\cos(x_3))\{u_1 - u_2 + J_6(x_2 + x_4)^2 \sin(x_3)\} \quad} \tag{4.10}$$

where

J_1 = mass moment of inertia of the link 1 w.r.t. the 1st axis

J_2 = mass moment of inertia of the link 2 w.r.t. the 2nd axis

J_3 = mass moment of inertia of the hand and the load w.r.t. the position of the hand.

$J_4 = J_2 + l_2^2 m_3$

$J_5 = J_1 + l_1^2(m_2 + m_3)$

$J_6 = l_1(r_2 m_2 + l_2 m_3)$

$J_7 = J_3 + J_4$

m_2 = mass of the second link

m_3 = mass of the hand and the load

l_1, l_2 = lengths of the links 1 and 2, respectively

r_2 = distance between the center of gravity of the second link and its driving axis

The control constraints are

$$|u_1(t)| \le M_{\varphi,max} \tag{4.11}$$

$$|u_2(t)| \le M_{\psi,max} \tag{4.12}$$

For the simulation the following values are used.

$J_1 = 1.6 m^2 kg$ $\quad J_2 = 0.43 m^2 kg$ $\quad J_3 = 0.01 m^2 kg$

$l_1 = 0.4 m$ $\quad l_2 = 0.25 m$ $\quad r_2 = 0.125 m$

$m_2 = 15 kg$ $\quad m_3 = 6 kg$ $\quad M_{\varphi,max} = 25 Nm$ $\quad M_{\psi,max} = 9 Nm$

4.2 *Time-Optimal Control Problem*

The cost function is

$$J(u) = \int_0^T dt \tag{4.13}$$

The Hamiltonian function is

$$H = 1 + p_1(t)\dot{x}_1(t) + p_2(t)\dot{x}_2(t) + p_3(t)\dot{x}_3(t) + p_4(t)\dot{x}_4(t) \quad (4.14)$$

It is of the form

$$H = f(x,p) + h_1(x,p)u_1 + h_2(x,p)u_2 \quad (4.15)$$

where

$p_i(t), i = 1,\cdots,4$ are costate variables

$$h_1(x,p) = \frac{0.815p_2 - (0.815 + 1.35\cos x_3)p_4}{4.0424 - 1.8225\cos^2 x_3} \quad (4.16)$$

$$h_2(x,p) = \frac{-(0.815 + 1.35\cos x_3)p_2 + (5.775 + 2.7\cos x_3)p_4}{4.0424 - 1.8225\cos^2 x_3} \quad (4.17)$$

The costate variables satisfy the linear adjoint differential equations

$$\dot{p}_i(t) = -\partial H / \partial x_i \qquad \text{for } i = 1\cdots 4 \quad (4.18)$$

Their boundary values at $t=0$ and $t=T$ are free because the initial and final states are fixed.

Using switching functions and applying Pontryagin's minimum principle we can get control laws. We see that except singular cases it becomes BangBang control.

4.3 *Solutions*

The solution will be discussed for the case where the robot arm is stretched both in the initial and in the final position.

$$x_1(0) = \varphi_a = 0 \quad (4.19)$$

$$x_3(0) = \psi_a = 0 \quad (4.20)$$

$$x_1(T) = \varphi_b > 0 \quad (4.21)$$

$$x_3(T) = \psi_b = 0, \pm 2\pi, \cdots \quad (4.22)$$

$$x_2(0) = x_4(0) = x_2(T) = x_4(T) = 0 \quad (4.23)$$

For $\varphi_b < 0$ the solution is symmetric so, the interesting region of motion is $0 \le \varphi_b \le \pi$. In solving this problem the shooting algorithm and trial and error methods were used.

There are 4 types of solution as followings.

Type 1 : $\psi_b = 0$

For φ_b(0 ~ 0.98 rad) , switching sequence for u_1 = (*max,min*) , u_2 = (*min,max,min*) .

Where the optimal torque switch sequence is abridged.

Type 2 : $\psi_b = 0$

For φ_b(0.98 ~ π rad) , switching sequences for u_1 = (*max,min,max*) , u_2 = (*min,max,min*)

Type 3 : $\psi_b = -2\pi$

For φ_b(0 ~ 1.76 rad) , switching sequence for u_1 = (*max,min,max,min*) , u_2 = (*min,max*)

Type 4 : $\psi_b = -2\pi$

For φ_b(1.76 ~ π rad) , switching sequences for u_1 = (*max,min*) , u_2 = (*min,max,min,max*)

Comparing the numerical values of the types for $0 \le \varphi_b \le \pi$, we can find global time-optimal solution.

Optimal Solution :

for φ_b(0 ~ 0.76 rad) , *Type 1*

for φ_b(0.76 ~ 1.76 rad) , *Type 3*

for φ_b(1.76 ~ π rad) , *Type 4*

Fig. 1. ~ 4. show examples of solutions respectively using plots for stroboscopic manipulator images. And phase plane trajectories are shown in Fig. 5. ~ 6.

The solution by global search method is not presented here and it needs a lot of computation. As stated earlier the method finds the good paths at first, and after local optimization which is path-constrained time-optimal motion problem it finds the best solution.

Obviously the solutions we have shown is globally optimal because we got them by solving the TPBVP directly.

So we can conclude that solutions by global search will be the same as those we have shown.

5. Conclusions

At first, global search method has been presented for computing the optimal motion of a manipulator, considering its dynamics, actuator constraints, joint limits, and obstacles. It is computationally efficient relative to other methods such as a dynamic programming search in the state space. But global search method requires relatively long time in performing local optimization which requires computing of path-constrained time-optimal motion. Then, path-constrained motion has been presented.

We can get the global optimal solution by solving TPBVP directly in a certain case. Example of 2-link manipulator case has been solved and optimal solution was acquired. We could see the validity of global search method intuitively.

For further study area we can say that a simpler and computationally efficient algorithm is needed. And the problem for the case of multi-robot that work together need to be studied.

References

[1] Z. Shiller and S. Dubowsky, "On computing the global time-optimal motions of robotic manipulators in the presence of obstacles" *IEEE Trans. Robotics Automat.*, vol. 7, no. 6, pp. 785-797, Dec. 1991.

[2] H. Geering, L. Guzzella, S. A. R. Hepner, and C. H. Onder, "Time-optimal motions of robots in assembly tasks," *IEEE Trans. Automat. Contr.*, vol. AC-31, no. 6, pp. 512-518, June 1986.

[3] T. Lozano-Perez, "A simple motion planning algorithm for general robot manipulators," *IEEE J. Robotics Automat.*, vol. RA-3, no. 3, pp. 224-238, June 1987.

[4] Z. Shiller and S. Dubowsky, "Time optimal path planning for robotic manipulators with obstacles, actuator, gripper and payload, constraints," *Int. J. Robotics Res.*, pp. 3-18, Dec. 1989.

[5] Z. Shiller, and H. H. Lu, "Robust computation of path constrained time optimal motions," in *Proc. IEEE Conf. Robotics Automat.* (Cincinnati, OH), May 1990, pp. 144-149.

[6] K. G. Shin and N. D. Mckay, "Minimum-time control of robotic manipulators with geometric path constraints," *IEEE Trans. Automat. Contr.*, vol. AC-30, no. 6, pp. 531-541, June 1985.

[7] Frank L. Lewiš, *Optimal control*, John Wiley & Sons, 1986

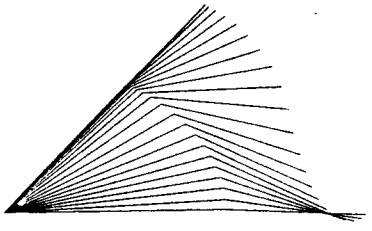

Fig. 1. *Type 1*, switching time for u_1 : 0.5425 sec, switching time for u_2 : 0.088, 0.588 sec, T = 1.085 sec, ψ_b = 0.979

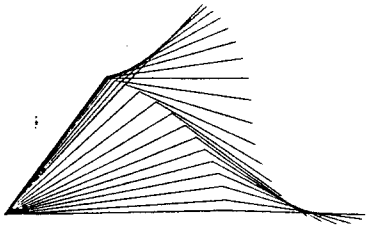

Fig. 2. *Type 2*, switching time for u_1 : 0.513, 1.056 sec, switching time for u_2 : 0.113, 0.634 sec, T = 1.085 sec, ψ_b = 0.981

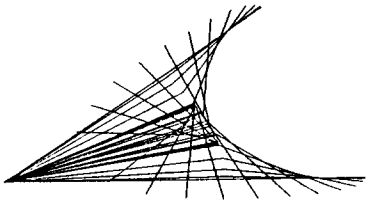

Fig. 3. *Type 3*, switching time for u_1 : 0.190, $T/2$, 0.785 sec, switching time for u_2 : $T/2$ sec, T = 0.9755 sec, ψ_b = 0.76

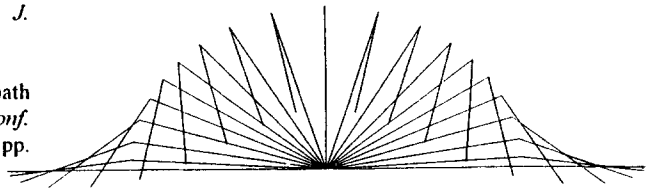

Fig. 4. *Type 4*, switching time for u_1 : $T/2$ sec, switching time for u_2 : 0.278, $T/2$, 0.974 sec, T = 1.252 sec, $\psi_b = \pi$

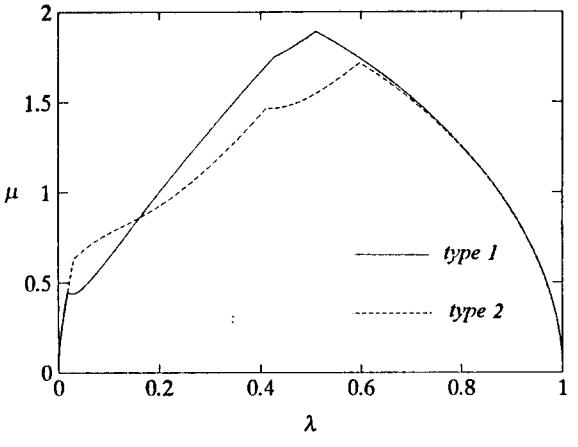

Fig. 5. Phase plane trajectories for *type 1* and *type 2*. where $0 \le \lambda \le 1$, $\mu = \dot{\lambda}$

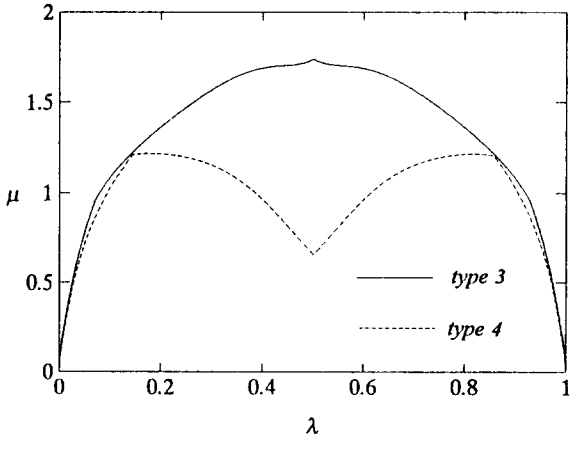

Fig. 6. Phase plane trajectories for *type 3* and *type 4*. where $0 \le \lambda \le 1$, $\mu = \dot{\lambda}$