INTERNATIONAL JOURNAL OF ROBOTICS
(THEORY AND APPLICATIONS)
E-mail: IJR@kntu.ac.ir

Near-Minimum-Time Motion Planning of Manipulators along Specified Path

M. J. Sadigh [a], M. H. Ghasemi [b]

a- Isfahan University of Technology, Isfahan, Iran;
b-Babol University of Technology, Babol, Iran.
Corresponding Author: Mohammad J. Sadigh, Isfahan University of Technology, Isfahan, Mechanical Engineering Department Phone: (98311)(3915206) Fax: (98311)(3912628), Jafars@cc.iut.ac.ir.

ARTICLE INFO
Keywords:
Optimal Path Planning
Parallel Manipulators
Switching Location

ABSTRACT
The large amount of computation necessary for obtaining time optimal solution for moving a manipulator on specified path has made it impossible to introduce an on line time optimal control algorithm. Most of this computational burden is due to calculation of switching points. In this paper a learning algorithm is proposed for finding the switching points. The method, which can be used for both serial and parallel manipulators, is based on a two-switch algorithm with three segments of moving with maximum acceleration, constant velocity and maximum deceleration along the path. The learning algorithm is aimed at decreasing the length of constant velocity segment in each step of learning process. The algorithm is applied to find the near minimum time solution of a parallel manipulator along a specified path. The results prove versatility of the algorithm both in tracking accuracy and short training process.

1. Introduction

Time optimal solution has always been an interesting subject among researches working on path planning and control of manipulators.

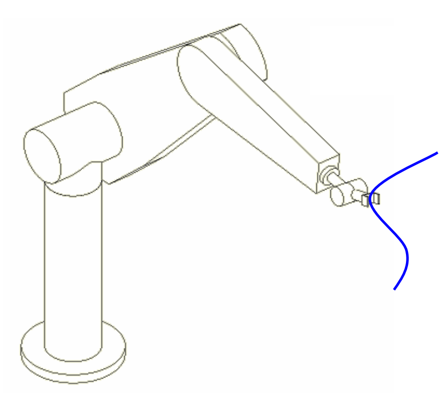

Figure 1. schematic diagram of the manipulator

The minimum time problem of tracking specified path by a serial manipulator was extensively studied by many researchers. Bobrow et al. [1] proposed a method for time optimal motion of serial manipulators based on phase plane analysis. Considering that the solution is bang-bang in terms of acceleration along the path, the method reduces the problem into calculating the maximum and minimum acceleration along the trajectory in each step, and to find the switching points. They used a geometric approach in the phase plane and suggested a shooting method for finding switching points, in which one has to find a solution trajectory which comes in contact with the boundary of non-feasible region (NFR) without crossing it, where NFR is part of phase plane in which no solution that keeps end effector on prescribed trajectory is feasible. This procedure is numerically very difficult and expensive task to do.

Their method was further developed by Pfeiffer and Johanni [2]. Taking advantage of characteristics of the boundary of Non-Feasible Region, they presented a method for direct calculation of this boundary and finding the switching points on it for serial manipulators. They stated that switching points might occur on the boundary of NFR at critical points, where the slope of non-feasible region boundary minus the value of $\ddot{s}/\dot{s}$

changes signs. This advancement considerably reduced the numerical effort.

Zlajpah [3] introduced the concept of trapped area from which no solution trajectory can escape without leaving the prescribed trajectory, and locked area to which no solution trajectory can enter from within feasible area.

Timar *et al.* [4] applied such methods to determine time-optimal solution for a CNC machine subject to prescribe acceleration bounds along axis. Sadigh and Hassan Ghasemi [5] showed that the lower boundary of these trapped and locked areas constructs the switching curve. The switching curve is a solution trajectory itself, and could be generated by direct integration of equations of motion, provided that either the first switch or one of the critical points on it is known.

The problem of minimum time motion along specified path for cooperative manipulators was also studied by several researchers. McCarthy and Bobrow [6] proved that for a manipulator with *n* coordinate, *p* differential constraint equations and *m* actuators, at least *m-n+p+1* actuators are saturated during a time optimal movement along a prescribed path. Taking advantage of this result, Sadigh and Hasan Ghasemi [5] proposed a direct method for calculation of maximum and minimum acceleration for CMMS. Moon and Ahmad [7] employed a similar algorithm as Bobrow et al. to find the time-optimal trajectory for a cooperative robot. They showed that to find the maximum and minimum values of acceleration at each point, one should solve a linear programming problem. They, however, did not elaborate on calculation of switching points, which itself is a very difficult part of the solution. Hasan Ghasemi and Sadigh [8] extended the work by Pfeiffer and Johanni to propose a direct method for computation of critical points for parallel manipulators and presented an algorithm to construct the switching curve. These advancements have made the situation for parallel manipulators similar to serial ones.

Minimum time motion of redundant manipulators along specified path is another interesting subject which has been studied in past two decades. Ma and Watanabe [9] and Galicki [10] extended the method proposed by Bobrow et.al. for serial redundant manipulators. They applied different secondary constraints such as heat characteristics and kinematic constraints to solve the problem.

Mattmüller and Gisler [11] presented a near time-optimal and jerk-constrained trajectory planner so that the constraint on the jerk translates into limits for the curvature of the phase-space velocity. Constantinescu and Croft [12] extended the time-optimal method to solve this problem with jerks constraints (limits on the rate of actuator torques) using perturbation trajectory improvement algorithm. Bianco and Piazzi [13] presented a global optimization approach to obtain a minimum-time cubic spline trajectory for manipulator point-to-multipoint operations subject to constraints given by limited joint torques and torque derivatives.

Osumi *et al.* [14] developed a time-optimal control method for quadruped walking robots and installed into a practical robot system. Yi *et al.* [14] presented a programming method for time-optimal control of a mobile robot.

In spite of all above mentioned advancements in this area during last two decades, which made it possible to compute the maximum and minimum acceleration on line, switching points needs off line computation. This fact, which is due to computation of critical points, and backward integration for first and last switching points, prevents this method to be used as a control algorithm. So far the method can only be used for time optimal path planning.

This paper takes advantage of the previous theoretical developments in this area and presents a learning algorithm to find switching points and near-minimum time solution. The method can be used both for serial and parallel manipulators. Considering the fact that minimum time solution is bang-bang in terms of tangential acceleration, the basic idea behind the proposed method is to move the manipulator on the specified path on consequent segments of maximum acceleration, constant velocity, and maximum deceleration and to learn the manipulator to reduce and adjust the constant velocity period in each step of learning process. As the constant velocity period gets smaller and smaller, the solution converges the time optimal and two switches on the start and final time of constant velocity period converge the real switch. Adjustment of second switch also pushes the final error to zero. In fact, this method does not propose a new optimization algorithm, but it is a method to reduce calculation effort and backward integrations, necessary for finding switching points during a minimum time motion. This method substitutes the tedious numerical procedure of calculation of switching points with a simple learning process. As a result of this reduction in numerical effort, the method could be used online in practical problems. After this introduction, a brief statement of time-optimal problem along with phase-plane solution given by Dubowskey [1] is presented in section two. The main idea of algorithm for single switch cases is discussed in third section. The fourth section is devoted to multi-switch cases followed by some numerical examples in fifth section.

2. Time Optimal Problem

Consider a non-redundant serial manipulator, which is supposed to move a payload from an initial point to a final point on a specified task space trajectory in minimum time subject to actuator's saturation limit. Motion of the payload in task space is defined by *n* coordinates; $\mathbf{X}=[X_1,...,X_n]$ and the motion of the system is defined with *n* variables; i.e. $\mathbf{q}=[q_1,...,q_n]^T$.

The equations of motion of such system can be written as

$$\mathbf{M}_{n\times n}(\mathbf{q})\,\ddot{\mathbf{q}}_{n\times 1}+\mathbf{h}_{n\times 1}(\mathbf{q},\dot{\mathbf{q}})=\mathbf{B}_{n\times m}(\mathbf{q})\boldsymbol{\tau}_{m\times 1} \qquad (1)$$

In this equation, $\mathbf{M}$, $\mathbf{h}$, and $\boldsymbol{\tau}$ are, respectively, the generalized mass matrix, coriolis and centrifugal terms, and the array of actuator forces.

The path in task space, $\mathbf{X}$, can be stated as

$$\begin{aligned}\mathbf{X} &= \mathbf{p}(\mathbf{q})\\ \dot{\mathbf{X}} &= \mathbf{J}(\mathbf{q})\dot{\mathbf{q}}\\ \ddot{\mathbf{X}} &= \mathbf{J}(\mathbf{q})\ddot{\mathbf{q}} + \dot{\mathbf{J}}(\mathbf{q})\dot{\mathbf{q}}\end{aligned} \tag{2}$$

Where $\mathbf{q}$, $\mathbf{p}$ and $\mathbf{J}$, respectively denote the array of joint coordinates, direct kinematic relation of the manipulator and its Jacobian. On the other hand, the path can be expressed in terms of the non dimensional arc length variable, s, as

$$\begin{aligned}\mathbf{X} &= \mathbf{f}(s)\\ \dot{\mathbf{X}} &= \mathbf{f}'(s)\dot{s}\\ \ddot{\mathbf{X}} &= \mathbf{f}'(s)\ddot{s} + \mathbf{f}''(s)\dot{s}^2\end{aligned} \tag{3}$$

In above equations, **f** shows the relation of the path in task space with non dimensional arc length parameter, s, also, (.)' denotes derivative with respect to s. Substituting for $\mathbf{X}$, $\dot{\mathbf{X}}$ and $\ddot{\mathbf{X}}$ from equations (3) into equations (2) and solving for $\mathbf{q}$, $\dot{\mathbf{q}}$ and $\ddot{\mathbf{q}}$ one gets:

$$\mathbf{q} = \mathbf{p}^{-1}(\mathbf{f}(s)) \tag{4}$$

$$\dot{\mathbf{q}} = \mathbf{J}^{-1}\dot{\mathbf{X}} = \mathbf{J}^{-1}\mathbf{f}'(s)\dot{s} \tag{5}$$

$$\ddot{\mathbf{q}} = \mathbf{J}^{-1}\ddot{\mathbf{X}} - \mathbf{J}^{-1}\dot{\mathbf{J}}\mathbf{J}^{-1}\dot{\mathbf{X}} = \mathbf{J}^{-1}(\mathbf{f}'(s)\ddot{s} + \mathbf{f}''(s)\dot{s}^2) - \mathbf{J}^{-1}\dot{\mathbf{J}}\mathbf{J}^{-1}\mathbf{f}'(s)\dot{s} \tag{6}$$

Where $\mathbf{P}^{-1}$ represent the inverse kinematics. Substituting equations (4), (5) and (6) into equation (2), one can rewrite equations of motion as:

$$\mathbf{c}_{n\times1}\ddot{s} + \mathbf{d}_{n\times1}\dot{s}^2 + \mathbf{e}_{n\times1} = \tilde{\mathbf{B}}_{n\times m}\boldsymbol{\tau}_{m\times1} \tag{7}$$

The above system of equations represent n equations with two states, $[s,\dot{s}]$ a similar formulation for a non-redundant parallel manipulator will result in equations of motion similar to equation (7); detailed formulation for such systems can be found in [8]. Any motion of the system, which moves the object on the prescribed path, must satisfy all above equations. Now, the optimization problem can be stated as:

Problem (1): Find the desired path, $s^*(t)$, which minimizes $\int_{t_0}^{t_f} dt$ subject to

$$\begin{aligned}&\mathbf{c}(s)\ddot{s} + \mathbf{d}(s)\dot{s}^2 + \mathbf{e}(s) = \tilde{\mathbf{B}}(s)\boldsymbol{\tau}\\ &\tau_{i,\min} \le \tau_i \le \tau_{i,\max} \qquad i = 1,\dots,m\end{aligned} \tag{8}$$

It can be shown, see Bobrow et al. [1], that the solution to this problem is bang-bang in $\ddot{s}$. To solve problem (1), one has to take the following steps:

1- Find the minimum and maximum acceleration at each step

2- Find the switching points and switch the acceleration at switching points.

It can be shown that all switching points are located on a switching curve [8], which for a specified manipulator is only a function of the desired trajectory. As stated in previous section there is no method for online calculation of switching points. In next section, we describe the proposed learning algorithm to find the switching points.

3. Single Switch Algorithm

We assume that the described path is such that minimum time motion can take place with a single switch. As explained in first section, in the first step of learning algorithm we start the motion from s_0 with maximum acceleration. The acceleration is then switched to zero at s_1, see Figure 2, and motion is continued with constant velocity until the end effector reaches point s_2 along the path. At this point acceleration is switched to its minimum possible value and the motion is continued until the line $\dot{s} = 0$ is crossed. With this planned motion, one might expect s at final point to be different from desired one, s_f.

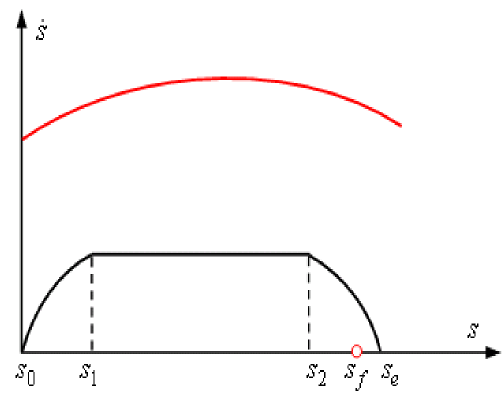

Figure 2. schematic diagram of first step in learning process

Considering that the minimum time solution is bang-bang in terms of $\ddot{s}$, we know that the final solution is obtained if s_2 coincide with s_1. In other words, the solution is obtained once constant velocity portion of motion is totally eliminated and the manipulator is either moving with maximum acceleration or deceleration along the path. With this in mind, we must suggest a learning algorithm which can decrease the distance between s_1^i and s_2^i and to decrease the final error, $\delta^i = s_e^i - s_f$. The first action causes two approximate switching points s_1 and s_2 to converge to the exact switch and second action causes the end effector to stop at the desired final point. To make the algorithm clear we first propose separate algorithms for these two actions and then try to combine them and present the final algorithm.

3.1 Elimination of Final Error

To eliminate the final error, assuming that minimum acceleration trajectories are almost parallel we may change the switching point s_2^i to $s_2^i - \delta^i$ at step *i+1* which means

$$s_2^{i+1} = s_2^i - \delta^i \tag{9}$$

This correction continues until the final error becomes smaller than a reasonable value, β. Figure 3 shows the algorithm of eliminating final error while s_1^i is kept constant.

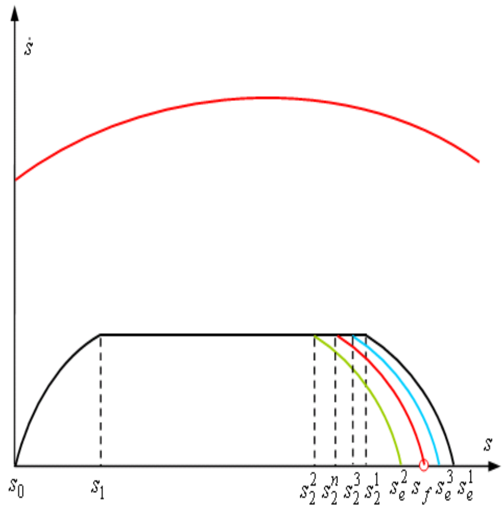

Figure 3. schematic diagram of the process eliminating final errors

3.2 Finding the Switching Point

To find the switching point, we increase s_1, and decrease s_2 at each step until these, two converge to the single switch. To this end, at each step we change the switching points by a value of ε as follow

$$s_1^{i+1} = s_1^i + \varepsilon^{i+1}$$
$$s_2^{i+1} = s_2^i - \varepsilon^{i+1} \qquad (10)$$

The value of ε^{i+1} can be considered as follow

$$\varepsilon^{i+1} = \alpha(s_f - s_0) \qquad if \qquad s_2^i - s_1^i > 2\alpha(s_f - s_0) \qquad (11)$$

In which, α is a constant which indicates how fast the switch points should approach each other in the early stages of the learning process, where the distance between s_1 and s_2 is still larger than twice of ε. The value of α should always be smaller than 0.5 and a good typical value for that would be some thing between 0.05 to 0.2. Smaller values of α means slower and safer approach to switch, while larger values of α means faster approach to the switch but at the risk of crossing non-feasible region boundary; i.e. leaving the desired trajectory. Figure 4.a shows schematic graph of this algorithm.

As the learning process advances and two switches approach each other, their distance become smaller than 2ε and it would not be possible to take the next step. In this case, considering the unsymmetric shape of the solution trajectory, see figure 4.b, we may assume that the distance of exact switch from s_1^i and s_2^i is different and is proportional to their distances travelled by maximum acceleration and maximum deceleration; i.e., s_1^i and $s_e^i - s_2^i$. With this assumption one might calculate ε^{i+1} as:

$$\varepsilon^{i+1} = \gamma \frac{s_1^i ds^i}{2(s_e^i - ds^i)} \qquad if \qquad s_2^i - s_1^i \le 2\alpha(s_f - s_0) \qquad (12)$$

In which γ is a constant which shows how fast the switch points should approach each other at final stages of the learning process. The maximum value for γ is one and a good typical value for γ would be something between 0.75 to 0.9.

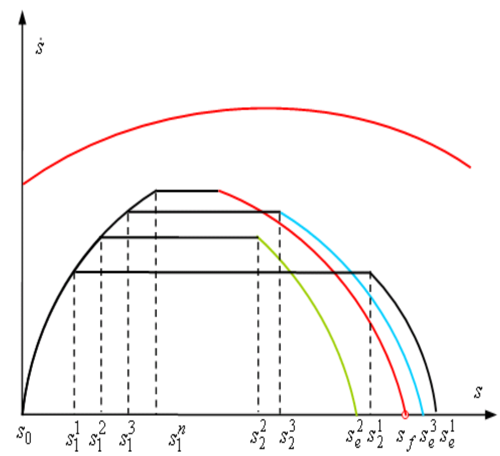

Figure 4.a. schematic diagram for algorithm of finding switching

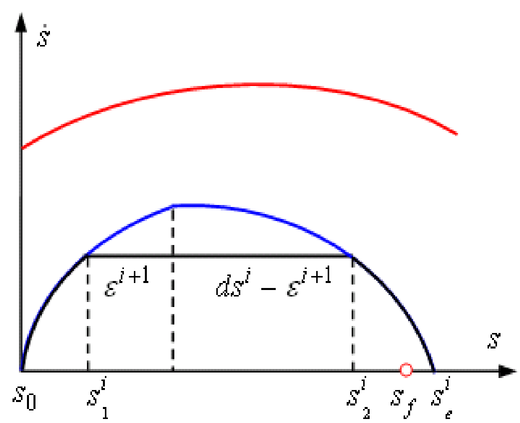

Figure 4.b. schematic diagram for new value of ε^{i+1}

3.3 Final Algorithm

At this stage, we may combine the above mentioned algorithms to obtain one which both approaches the approximate switches to the real one and to eliminate the final error. To this end, it is sufficient to make corrections to s_2^i based on both algorithms which means to calculate s_2^{i+1} as follow

$$s_2^{i+1} = s_2^{\,i} - \varepsilon^{i+1} - \delta^i \tag{13}$$

Where δ^i and ε^i are as defined in equation (9) to (12). Figure 5 shows how the final combined algorithm works.

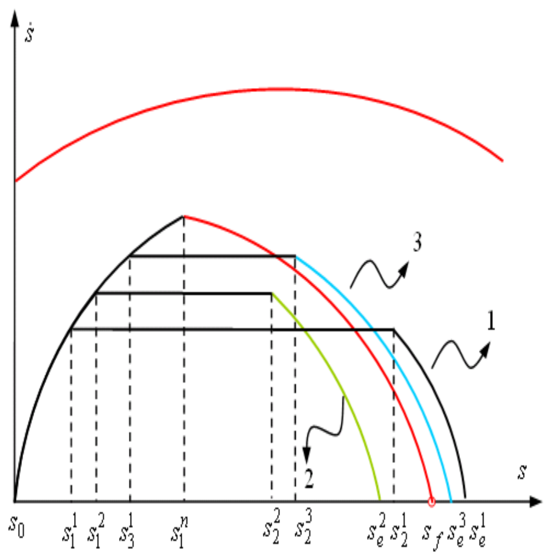

Figure 5. schematic diagram of results of minimum time algorithm

4 Multi Switch Algorithm

In this section, we consider the case where solution trajectory in phase plane enters non-feasible region; i.e. end effector leaves the prescribed path. For instance, suppose that at $i+1^{st}$ iteration, solution trajectory enters NFR, as shown in Figure 6. In this case, in next learning iteration we simply try to perform the single switch algorithm once between points (s_1^i, $\dot{s}^i$) and (s_{c1}, $\dot{s}^i$), and then between points (s_{c1}, $\dot{s}^i$) and (s_2^i, $\dot{s}^i$), see Figure 6. If in the process of finding these switches, the solution trajectory again intersects the NFR, a second critical point s_{c2} is introduced and similar algorithm of single switch is applied for that. To ensure escaping from crossing the NFR again and again it is suggested that ε^{i+1} be reduced effectively. This means that in neighbourhood of a difficult portion of the path increase of velocity must be slowly.

5 Numerical Example

Figure 7 shows the schematic of a system composed of two planar manipulators handling a payload. The physical characteristics of the system are indicated in Table 1. Each manipulator has three DOFS. It is assumed that the payload is rigidly grasped such that no slipping or rotation is possible at contact points.

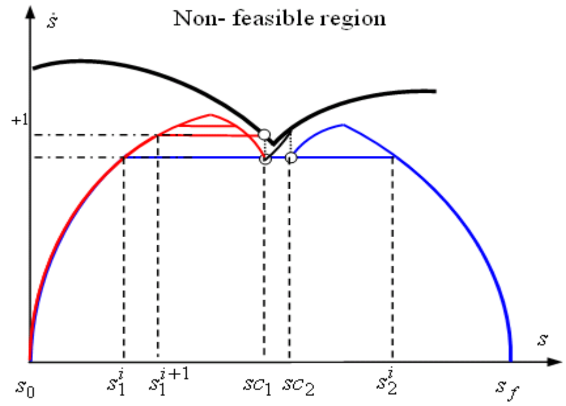

Figure 6. schematic diagram of multi switch algorithm

Table 1. physical characteristic of the system

$L_1 = L_4 = 0.5\,m,\ L_2 = L_5 = 0.6\,m,\ L_3 = L_6 = 0.3\,m,\ L_o = 0.2\,m$
$m_1 = m_4 = 1\,kg\quad m_2 = m_5 = 1\,kg\quad m_3 = m_6 = 0.3\,kg\quad m_o = 1\,kg$
$\boldsymbol{\tau}_{\max} = [70, 50, 20, 70, 50, 20]^T\ N.m\ ,\quad \boldsymbol{\tau}_{\min} = -\boldsymbol{\tau}_{\max}$
$b_0 = 0.7\,m$

5.1 Single Switch Problem

The system is assumed to move the payload on a pre-scribed path defined by equation (14), see Figure 7.

$$x(s) = 0.3s$$
$$y(s) = 1.02 - 0.7s \qquad 0 \le s \le 1 \tag{14}$$
$$\varphi(s) = \frac{\pi}{6}s$$

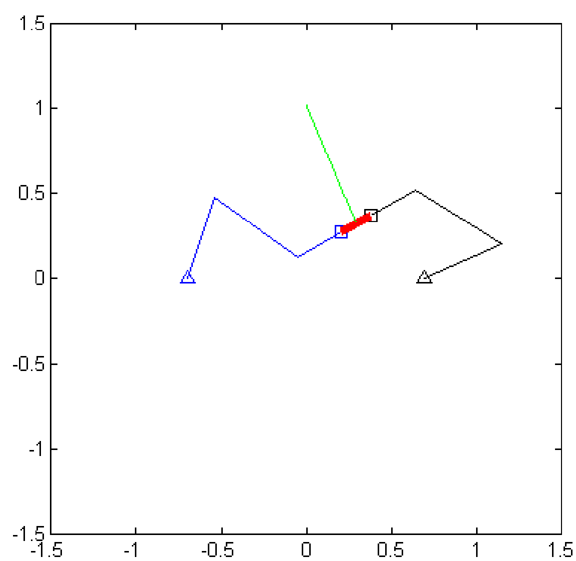

Figure 7. specified path and final configuration

Exact solution obtained by forward integration with maximum acceleration from initial point and backward integration from final point results in 0.4693 as switching point and 0.1647 as the total time elapse. This problem is solved taking advantage of the proposed method. The constant α is considered to be 0.1 which means s_1^1 is taken as 0.1 and s_2^1 as 0.9. As one can see after four steps the conditions of $s_2^i - s_1^i > 2\alpha(s_f - s_0)$ is violated and ε^i is reduced from 0.1 to 0.047, and in next step to 0.0152. As the results in Table 2 show, approximate switches converge to the real switch after six steps of learning process. As can be seen this algorithm could obtain the switching time and the minimum time solution with no backward integration or length calculations. Figure 8 shows the final trajectory and boundary of non-feasible region for desired trajectory in phase plane.

Table 2. simulation results for single switch problem

Step	s_1^i	s_2^i	$ds^i = s_2^i - s_1^i$	$\dot{s}^i$	ε^i	δ^i	t^i
1	0.1000	0.9000	0.8000	6.3781	0.1000	0.0856	0.2188
2	0.2000	0.6815	0.4815	8.8294	0.1000	-0.0357	0.1698
3	0.3000	0.6239	0.3239	10.4794	0.1000	0.0168	0.1700
4	0.4000	0.5052	0.1052	11.6739	0.1000	-0.0209	0.1628
5	0.4470	0.4798	0.0328	12.0865	0.0470	-0.0117	0.1637
6	0.4622	0.4765	0.0143	12.2121	0.0152	-0.0044	0.1647

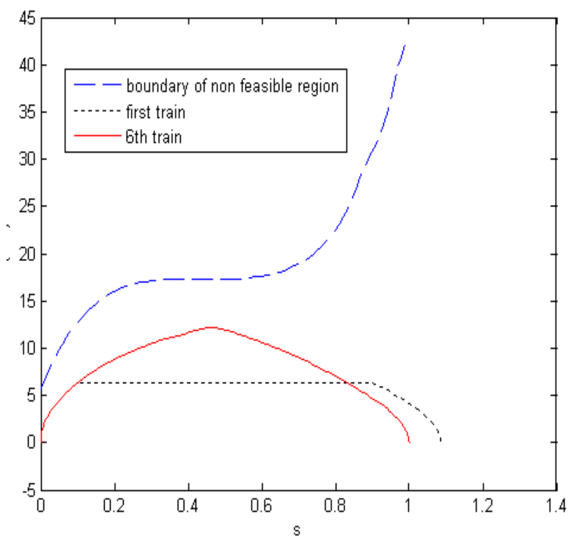

Figure 8. simulation results of learning process in phase plane

5.2 Multi Switch Problem

The system is assumed to move the payload on a circular path defined by equation (15), see Figure 9.

$$x(s) = 0.2\cos(\pi s + \frac{2\pi}{3})$$
$$y(s) = 0.2\sin(\pi s + \frac{2\pi}{3}) + 0.62 \qquad 0 \le s \le 1 \qquad (15)$$
$$\varphi(s) = 0.5s$$

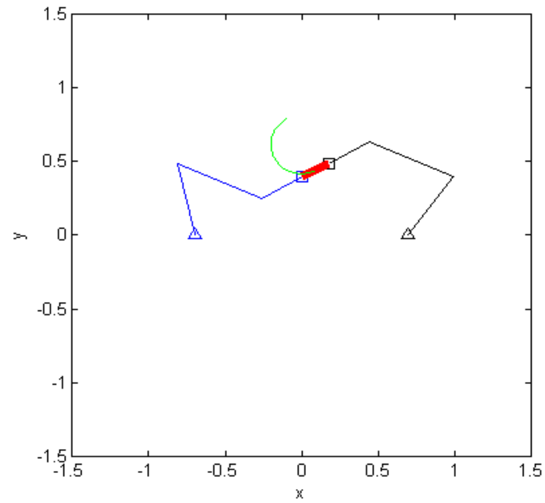

Figure 9. specified path and final configuration

Solution of this problem also starts based on the single switch algorithm stated in section 3 with $\alpha = 0.08$. As one can see from Table 3, after two steps, the solution trajectory intersected the boundary of non-feasible region. As suggested in section4, we tried to find one switch before s_{c1} and one switch after s_{c1}. Simulation results for this process are given in Table 4. As one can see after 30 steps of learning process both switches on left and right of s_{c1} are converged. However, learning process for finding these switches are very costly. Another point which worth mentioning is that the results obtained in first learning step is 10% more than the time elapse obtained from exact optimal solution of the problem, which is equal to 0.1981 sec. the next 30 steps of learning has reduced this 10% error to 1.3%. Figure 10 shows the final trajectory in phase plane as well as the non-feasible region boundary.

Exact solution to this problem by the conventional technique introduced by Bobrow et.al. [1] amounts to calculation of NFR boundary which needs a point to point calculation for each value of s from zero to one. Then the critical points on this boundary are to be calculated based on the algorithm introduced by Pfeiffer and Johanni [2] and Ghasemi and Sadigh [8]. Then one needs to perform direct integration from initial point and from critical points with maximum acceleration and backward integration from final point and critical points with maximum deceleration. The final step in the solution would be to intersect generated trajectories to obtain exact switches. Comparing the simple calculations needed for comparing switching points in proposed algorithm with the lengthy and time consuming computations necessary for obtaining exact switching points reveals the significance of proposed technique.

Table 3. numerical results for circular path before crossing BNF

step	s_1^i	s_2^i	$ds^i = s_2^i - s_1^i$	$\dot{s}^i$	ε^i	δ^i	contact	t^i
1	0.1000	0.9000	0.8000	5.6762	0.0800	0.0122	1	0.2182
2	0.1800	0.8032	0.6232	7.8189	0.0800	0.0122	0	

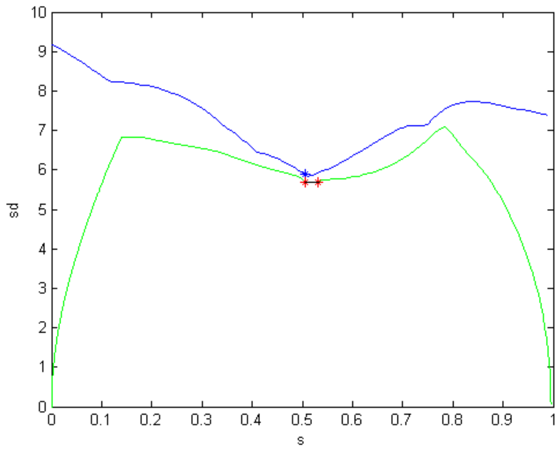

Figure 10. **boundary of non-feasible region and solution path in phase plane**

Table 4. numerical result of near minimum time motion

step	s^i_{a1}	s^i_{a2}	ds^i_a	$\dot{s}^i_a$	ε^i	s^i_{b1}	s^i_{b2}	ds^i_b	$\dot{s}^i_b$	t^i
1	0.1058	0.3998	0.2940	5.8781	0.0050	0.5406	0.9900	0.4494	5.7351	0.2278
5	0.1168	0.3763	0.2595	6.1843	0.0050	0.7406	0.8028	0.0622	6.6891	0.2019
10	0.1328	0.4107	0.2779	6.6381	0.0050	0.7656	0.7977	0.0321	6.9572	0.196
15	0.1398	0.1767	0.0369	6.8417	0.0050	0.7856	0.7860	0.0004	7.0930	0.2013
20	0.1403	0.1792	0.0389	6.8515	0.0050	0.7856	0.7860	0.0004	7.0930	0.2011
25	0.1408	0.1817	0.0409	6.9005	0.0050	0.7856	0.7860	0.0004	7.0930	0.2009
30	0.1413	0.1842	0.0429	6.9005	0.0050	0.7856	0.7860	0.0004	7.0930	0.2007

6. Conclusion

Problem of on line computation of switching points for a time optimal problem of a manipulator moving along a specified path is considered. The procedure can be used for on line evaluation of open loop time optimal control for both serial and parallel manipulators moving on a prescribed path. The method is based on the idea of moving end effector on the specified path on consequent segments of maximum acceleration, constant velocity, and maximum deceleration, and to learn the control to reduce and adjust the constant speed interval at each step of the learning process. This way two switches finally converge to the exact one and the final error is also eliminated. A development of the algorithm is also given for multi switch cases. The validity of the method is checked by solving time optimal problem for two cases of a double three link planar parallel manipulator, along a straight line and then along a circular line. The results for straight line show that in six steps of training, the final error is less than 0.44% and the travelling time is 0.28% more than the exact minimum time. These results are very promising both in accurate tracking and in fast learning process. Similar results are also reported for the circular path.

References

[1] Bobrow, J. E., Dubowsky, S. and Gibson, J.S.: *Time-optimal control of robotic manipulators along specified paths*, Int. J. Robotics Res., 1985, vol.4, no.3, pp.3-17.

[2] Pfeiffer, F. and Johanni, R.: *A Concept for Manipulator Trajectory Planning*, IEEE Journal of Robotics and Automation, vol.RA-3, 1987, no.2, pp.115-123.

[3] Zlajpah, L.: *On Time Optimal Path Control of Manipulators with Bounded Joint Velocities and Torques*, In Proc of IEEE Int. Conf. on Robotics and Automation, Minneapolis, 1996, pp.1572 - 1577.

[4] S. D. Timar, R.T. Farouki, and T.S. Smith, C.L. Boyadjieff, Algorithms for time-optimal control of CNC machines along curved tool paths, *Robotics and Computer-Integrated Manufacturing*, 2005, 21, 37–53.

[5] Sadigh, M. J. and Ghasemi, M.H.: *A Fast Algo rithm for Time Optimal Control of a Cooperative Multi Manipulator System on Specified Path*, in proc of 5th Vienna symposium on mathematical modeling, modeling for/and control, 2006, vol.2, pp.1-7.

[6] McCarthy, J. M. and Bobrow, J. E.: *The Number of saturated Actuators and Constraint Forces During Time-Optimal Movement of a General Robotic System*, IEEE Transaction on Robotics and Automation, 1992, vol.8, no.3, pp.407-409, June.

[7] Moon, S. B. and Ahmad, S.: *Time-Optimal Trajectories for Cooperative Multi-Manipulator System*, IEEE Transaction on system, Man and cybernetics, 1997, vol.27, no.2, pp.343-353.

[8] Ghasemi, M. H. and Sadigh, M. J.: *A Direct Algorithm to Compute Switching Curve for Time Optimal Motion of Cooperative Multi-Manipulators Moving on Specified Path*, International Journal of Advanced Robotics, 2008, vol.22, no.5, pp.493-506.

[9] S. Ma and M. Watanabe, Time-optimal control of kinematically redundant manipulators with limit heat charac teristics of actuators, Advanced Robotics, 2002, vol.16, no.8, pp.735 – 749

[10] M. Galicki, Control of kinematically redundant manipulator with actuator constraints, Robot Motion and Control,in proc of the Fifth International Workshop (RoMoCo '05), 2005, pp.123- 130.

[11] J. Mattmüller and D. Gisler, Calculating a near time-optimal jerk-constrained trajectory along a specified smooth path, *International Journal of Advanced Manufacturing Technology*, 2009, 45, 1007–1016.

[12] D. Constantinescu and E. A. Croft, Smooth and time optimal trajectory planning for industrial manipulators along specified paths, *Journal of Robotic Systems*, 2000, 17, 233-249.

[13] Bianco, C. G. L. and Piazzi, **A.** Minimum-time trajectory planning of mechanical manipulators under dynamic constraints, *International Journal of Control*, 2002, **75**(13), 967-980.

[14] H. Osumi, S.Kamiya, H. Kato, K. Umeda, R. Ueda, and T. Arai, Time optimal control for quadruped walking robots, in *Proc. IEEE International Conference on Robotics and Automation*, 2006, Orlando, pp. 1050-4729.

[15] F. Y. Yi, K. CH. Nan, L. T. Li, and W. CH. Ju, A Nonlinear Programming Method for Time-Optimal Control of an Omni-Directional Mobile Robot. *Journal of Vibration and Control,* 2008, 14, 1729-1747.

Biography of Authors

Mohammad Jafar Sadigh received his B.S. and M.S. degrees in Mechanical engineering both from Isfahan University of Technology (IUT), Isfahan, Iran in 1986 and 1989, respectively, and PhD from McGill University Montreal, Canada in 1995. Since 1995 he has been with the Department of Mechanical engineering at Isfahan University (IUT), Isfahan, Iran. His research interests are system dynamics, control of dynamical systems, robotics, and also he is very active in the field of science park development and technology management.
(Email: jafars@cc.iut.ac.ir)

Mohammad Hasan Ghasemi received his B.S. degree in Mechanical Engineering in 1998 from iran University of science and technology (IUST), Tehran, Iran. M.S. degree and PhD in Mechanical **engineering** both from Isfahan University of Technology (IUT), Isfahan, Iran in 2000 and 2008, respectively. Since 2008 he has been with the Department of Mechanical Engineering at Babol University of Technology, Babol, Iran. His research interests are robotics and control of dynamical systems. (Email: mhghasemi@nit.ac.ir)