Proceedings of the ASME 2014 Dynamic Systems and Control Conference
DSCC2014
October 22-24, 2014, San Antonio, TX, USA

DSCC2014-6143

MINIMUM-TIME CORNERING FOR MANUFACTURING MACHINES USING OPTIMAL CONTROL

Molong Duan
Mechanical Engineering
University of Michigan
Ann Arbor, MI 48109-2125
molong@umich.edu

Chinedum E. Okwudire
Mechanical Engineering
University of Michigan
Ann Arbor, MI 48109-2125
okwudire@umich.edu

ABSTRACT

When traversing sharp corners, manufacturing machines are forced to tradeoff speed and accuracy. The most common way of reducing this tradeoff is to smooth the sharp corner using a pre-specified curve (e.g., a circular arc or spline). However, pre-specified curves cannot guarantee optimal performance. This paper presents a preliminary investigation into the potential of using methods from optimal control to minimize this tradeoff. First, a useful simplification is made to the exact cornering problem to make it tractable. Dynamic programming is then used to determine the best free-form curve that minimizes corner traversal time while adhering to path tolerance and machine kinematic constraints. Significant improvements in cornering time are demonstrated compared to two methods that use pre-specified curves. However, dynamic programming is found to be too computationally costly, thus impractical. Less computationally intensive techniques in optimal control are considered for future work.

1. INTRODUCTION

High throughput and accuracy are conflicting requirements that must be met by advanced manufacturing machines like laser cutters, 3D printers and CNC machine tools. This tradeoff is very frequently encountered when executing sharp corners where, to achieve tight tolerances, machine axes have to slow down considerably thus sacrificing cycle time. The most common way of reducing this trade-off is to smooth the corner using a pre-specified curve that allows high-speed cornering subject to path tolerance and machine kinematic limits. The simplest smoothing curve, commonly used in industry, is the circular arc. However, a lot of research has been done on other smoothing techniques, e.g., double clothoid curves [1], quintic splines [2], P-H curves [3] and Bezier curves [4]. Such pre-specified curves are however not guaranteed to minimize the tradeoff between speed and accuracy, thus they may limit the performance of advanced manufacturing machines. This paper explores a different approach, based on optimal control methods, where the smoothing curve does not have to be pre-specified. Section 2 presents the exact formulation of the minimum-time cornering problem, and in Section 3, a simplification is made to the exact formulation of the problem to make it tractable using optimal control methods. A dynamic programming (DP) algorithm for solving the simplified problem is briefly described in Section 4 and, in Section 5, the results from DP are compared to those obtained using two pre-specified curves – a circular arc and a Bezier curve [4] – followed by discussions, conclusions and future work.

2. EXACT FORMULATION OF MINIMUM-TIME CORNERING PROBLEM

Consider the planar cornering scenario shown in Fig. 1, arising from the intersection of two line segments generated by perpendicular axes (x and y) of a manufacturing machine. $\mathbf{p}_1 = \{0,0\}^T$ is the resulting sharp corner, while angles θ_1 and θ_2 respectively define the orientations of the incoming and outgoing line segments. Based on the unit vectors $\mathbf{t}_1 = \{\cos\theta_1, \sin\theta_1\}^T$ and $\mathbf{t}_2 = \{\cos\theta_2, \sin\theta_2\}^T$ along each line, two boundary points $\mathbf{p}_0 = \mathbf{p}_1 + l_1\mathbf{t}_1$ and $\mathbf{p}_2 = \mathbf{p}_1 + l_2\mathbf{t}_2$ are specified to mark the start and the end of the corner region surrounding $\mathbf{p}_1$. The exact corner path can therefore be defined, per Ernesto and Farouki [4], by the point set

$$\hat{\mathbf{r}} = \begin{cases} \mathbf{p}_0(1-2\xi) + \mathbf{p}_1(2\xi - 0) & \text{for } \xi \in \left[0, \frac{1}{2}\right] \\ \mathbf{p}_1(2-2\xi) + \mathbf{p}_2(2\xi - 1) & \text{for } \xi \in \left[\frac{1}{2}, 1\right] \end{cases} \quad (1)$$

For a manufacturing machine to perfectly negotiate the sharp corner defined in Eq.(1), it would have to come to a complete stop at $\mathbf{p}_1$ before continuing on to $\mathbf{p}_2$, thus severely sacrificing cornering speed. However, having a perfectly sharp corner is not often required in manufacturing processes. It is instead permissible for the actual path to deviate from the exact corner within specified tolerance constraints. Therefore, it is customary to modify the corner trajectory such that the machine traverses it as fast as possible while keeping to the speed and tolerance constraints of the manufacturing process, as well as the kinematic limits of the machine axes. Let this modified trajectory be defined by the point set

$$\mathbf{r}(t) = \{x(t), y(t)\}^T \quad \forall\, t \in [t_0, t_2] \tag{2}$$

where t_0 and t_2 are respectively the time instants at $\mathbf{p}_0$ and $\mathbf{p}_2$. To satisfy continuity requirements at $\mathbf{p}_0$ and $\mathbf{p}_2$, $\mathbf{r}(t)$ must satisfy the conditions

$$\mathbf{r}(t_0) = \mathbf{p}_0; \quad \mathbf{r}(t_2) = \mathbf{p}_2 \tag{3}$$

Similarly, $d\mathbf{r}(t)/dt$ must satisfy the end conditions

$$\frac{d\mathbf{r}(t_0)}{dt} = -V_0\mathbf{t}_1; \quad \frac{d\mathbf{r}(t_2)}{dt} = V_2\mathbf{t}_2 \tag{4}$$

where V_0 and V_2 are the speeds at $\mathbf{p}_0$ and $\mathbf{p}_2$, respectively. Given the maximum allowable corner error ε, the tolerance constraint on $\mathbf{r}(t)$ can be expressed [1] as

$$d_H(\mathbf{r}, \hat{\mathbf{r}}) \le \varepsilon \tag{5}$$

d_H is the Hausdorff distance defined as

$$d_H(\mathbf{r}, \hat{\mathbf{r}}) = \max\left(\sup_{\mathbf{a}\in\hat{\mathbf{r}}}\inf_{\mathbf{b}\in\mathbf{r}} d(\mathbf{a},\mathbf{b}), \sup_{\mathbf{b}\in\mathbf{r}}\inf_{\mathbf{a}\in\hat{\mathbf{r}}} d(\mathbf{a},\mathbf{b})\right) \tag{6}$$

where d is the Euclidean distance. The constraint of Eq.(5) implies that the modified corner trajectory must lie within the dashed lines surrounding the exact corner shown in Fig. 1. Additionally, the trajectory must satisfy the constraints

$$\left|\frac{d\mathbf{r}(t)}{dt}\right| \le V_{\max} \tag{7}$$

and

$$-\begin{Bmatrix} A_x \\ A_y \end{Bmatrix} \le \frac{d^2\mathbf{r}(t)}{dt^2} \le \begin{Bmatrix} A_x \\ A_y \end{Bmatrix} \tag{8}$$

$V_{\max}$ is the maximum allowable speed along the path (often dictated by the manufacturing process), while A_x and A_y are the acceleration/deceleration limits of the x and y axes, respectively. The optimal trajectory $\mathbf{r}(t)$ is the one that minimizes the objective function

$$J = \int_{t_0}^{t_2} dt \tag{9}$$

subject to the constraints defined in Eqs. (3) through (8). The solution of this optimization problem is extremely challenging, in large part due to the complicated point-wise geometric conditions imposed by the Hausdorff distance constraint, defined in Eq.(5), which requires the entire trajectory to be known beforehand. A simplification of the problem is therefore needed to facilitate its solution.

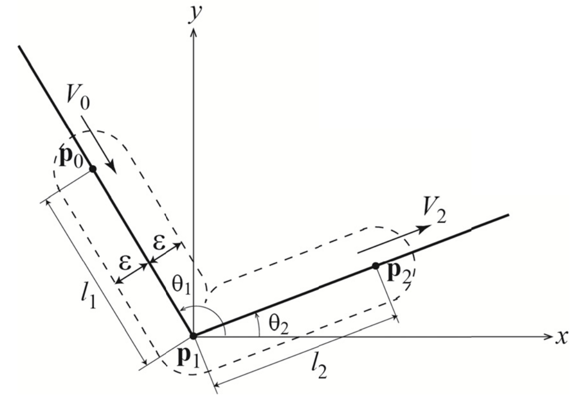

Figure 1. CORNER REGION OF TWO INTERSECTING LINES

3. SIMPLIFIED FORMULATION OF MINIMUM-TIME CORNERING PROBLEM

One approach to simplifying the optimal cornering problem defined above is to separate its geometric constraints (i.e., Eqs. (3) and (5)) from the kinematic constraints (i.e., Eqs. (4), (7) and (8)). The geometric shape of the modified trajectory is pre-defined using a parametric curve $\mathbf{r}(\xi)$ (where $\xi \in [0, 1]$ is the curve parameter) such that it satisfies the geometric constraints. With the path now fixed, the objective function can be reduced to

$$J = \int_{t_0}^{t_2} dt = \int_0^1 \left|\frac{d\mathbf{r}(\xi)}{d\xi}\right| \frac{1}{V(\xi)} d\xi \tag{10}$$

from which the optimal speed profile $V(\xi)$ that satisfies the established kinematic constraints is determined. This approach, generally known as optimal feed rate scheduling, is widely used in trajectory planning for robotics and manufacturing applications, e.g., [5, 6], and has recently been applied to the minimum-time cornering problem by Ernesto and Farouki [4].

While pre-defining the corner path greatly simplifies the optimization problem, it could lead to sub-optimal solutions because it does not guarantee that the selected curve is most favorable for the optimization. To address this limitation, a less rigid simplification to the optimal cornering problem is proposed in this paper. $\mathbf{p}_1$ is assumed to be the critical point along the trajectory; i.e., it is assumed that if the tolerance condition of Eq.(5) is satisfied at $\mathbf{p}_1$ then it can be made to be satisfied everywhere else on the point set $\mathbf{r}(t)$. Therefore, the position and slope of $\mathbf{r}(t)$ at $\mathbf{p}_1$ are selected such that

$$\mathbf{r}(t_1) = \{\varepsilon\cos\beta, \varepsilon\sin\beta\}^T; \quad \frac{\dot{\mathbf{r}}(t_1)}{|\dot{\mathbf{r}}(t_1)|} = dir \cdot \{\sin\beta, -\cos\beta\}^T \tag{11}$$

where $\beta = (\theta_1 + \theta_2)/2$ and $dir = \text{sgn}(\{\sin\beta, -\cos\beta\}^T\{\cos\theta_2, \sin\theta_2\})$. Eq. (11) implies that the velocity at the critical point must be perpendicular to the bisector of the corner, and be directed towards $\mathbf{p}_2$. This allows $\mathbf{r}(t)$ to be split into two portions, $\mathbf{r}_1(t)$ for $t \in [t_0, t_1]$ and $\mathbf{r}_2(t)$ for $t \in [t_1, t_2]$ whose geometries are not pre-determined but instead are allowed to vary according to the dictates of the optimization performed, as a preliminary step, using DP in this paper.

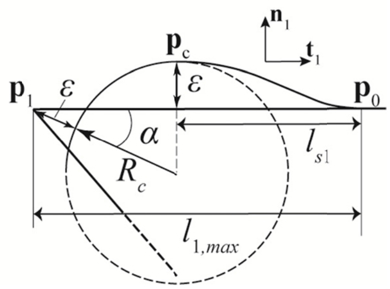

Figure 2. ESTIMATION OF $l_{i,max}$

We observe empirically that the length l_i (i = 1, 2) is critical for the assumption that $\mathbf{p}_1$ is the most critical point to be valid. The reason is that the optimization seeks to minimize the curvature along the corner path so that higher speeds can be achieved while keeping to the imposed acceleration and endpoint constraints. If l_i is small enough, the endpoint constraints combined with curvature restraints are sufficient to prevent the optimized path from violating the tolerance conditions. We therefore seek to provide an estimation of the maximum l_i (i.e., $l_{i,\max}$) that ensures that our assumption is valid for a given cornering problem. To do this, let us assume (see Fig. 2) that the optimized corner path consists of a circular arc from the point defined in Eq.(11) to a point $\mathbf{p}_c$ where the tolerance constraint of Eq.(5) is on the verge of violation. The radius R_c of the circular arc is expressed as

$$R_c = \varepsilon \cot^2\left(\frac{\pi}{4} - \frac{\alpha}{2}\right); \qquad \alpha \triangleq \frac{|\theta_1 - \theta_2|}{2} \tag{12}$$

R_c is assumed to be equal to the smallest radius of curvature along the optimized path. Accordingly, the maximum allowable constant speed V_c along the arc is given by

$$V_c = \sqrt{R_c A_c}; \qquad A_c \triangleq \min\left(A_x, A_y\right) \tag{13}$$

Note that if $V_c > V_{\max}$, R_c in Eq.(12) cannot be achieved, meaning that the tolerance constraint will not be violated for any value of l_i. However, if $V_c \le V_{\max}$, $l_{i,\max}$ exists. To determine $l_{1,\max}$, assume that $\mathbf{p}_c$ is connected to $\mathbf{p}_0$ by a path with uniform acceleration/deceleration along each motion axis. The distance l_{s1} in Fig. 2 must be long enough to allow a transition from velocity V_0 to V_c along the unit vector $\mathbf{t}_1$ without violating the acceleration limits along $\mathbf{t}_1$. It must also allow the motion normal to $\mathbf{t}_1$ to travel the distance ε from rest to rest while staying within the acceleration constraints normal to $\mathbf{t}_1$. This condition is expressed as

$$l_{s1} = \max\left(\frac{\left|V_c^2 - V_0^2\right|}{2A_{t1}}, \left(V_c + V_0\right)\sqrt{\frac{\varepsilon}{A_{n1}}}\right) \tag{14}$$

where A_{t1} and A_{n1} are the acceleration limits along and perpendicular to $\mathbf{t}_1$ given by

$$A_{t1} = \min\left(\frac{A_x}{|\cos\theta_1|}, \frac{A_y}{|\sin\theta_1|}\right); \; A_{n1} = \min\left(\frac{A_x}{|\sin\theta_1|}, \frac{A_y}{|\cos\theta_1|}\right) \tag{15}$$

Based on the foregoing considerations, $l_{1,\max}$ can be derived as

$$l_{1,\max} = l_{s1} + \left(R_c + \varepsilon\right)\cos\alpha \tag{16}$$

By making necessary modifications to Eqs.(14)-(16), $l_{2,\max}$ can be determined in a similar manner.

The simplifying assumptions that have been made in determining $l_{i,\max}$ have been chosen so that the resulting $l_{i,\max}$ is conservative. This ensures that the tolerance condition is not violated when it is employed in generating an optimal corner trajectory, as is demonstrated in Section 5 of this paper.

4. DP BASED OPTIMIZATION ALGORITHM

Dynamic programming (DP) is a well-established method in the field of optimal controls. It is based on the Principle of Optimality put forward by Bellman [7]. Optimal trajectories calculated using DP are often used as benchmarks for evaluating those obtained from other (approximate) methods. Hence we adopt DP in this preliminary study to determine the potential benefits of employing optimal control methods for reducing cornering time in advanced manufacturing machines.

Figure 3 shows the flowchart of the DP algorithm for solving the simplified minimum-time cornering problem presented in the preceding section. The algorithm starts by discretizing the 4-D state space (consisting of the two components of $\mathbf{r}(t)$ and $d\mathbf{r}(t)/dt$). The initial trajectory is assumed to be a circular arc that satisfies the geometric constraint defined in Eqs. (11). $\mathbf{r}(t)$ is discretized in Δl and Δn increments along and normal to the vector $\mathbf{t}_i$ (i = 1, 2). $d\mathbf{r}(t)/dt$ is discretized in increments of Δv along the x and y directions so that A_x and A_y can be easily computed. Next, a check is performed to ensure that the mesh sizes satisfy the following two criteria

$$\frac{\Delta v}{\Delta t_{\min}} \le \min\left(A_x, A_y\right) \tag{17}$$

$$\Delta v \cdot \Delta t_{\min} \approx \Delta n \tag{18}$$

where $\Delta t_{\min} = \Delta l / V_{\max}$ is the minimum time interval between two consecutive points on the mesh. Eq.(17) guarantees that the solution can transition between two consecutive points on the velocity mesh using the maximum acceleration. Eq.(18) ensures that a velocity increment of Δv on the velocity mesh is distinguishable on the position mesh so that the velocity mesh is not too fine relative to the position mesh. The algorithm progresses to the next step if both criteria are satisfied. Otherwise, the mesh is modified.

The next step involves searching for the optimal trajectory using the Principle of Optimality [6]. The search involves iteratively updating a 4-D matrix $\mathbf{T}$ of optimal cornering times according to the equation

$$\mathbf{T}\left(l_k, n_{p,k}, u_{q,k}, v_{r,k}\right) = \min_{p,q,r\in\Omega}\left\{\mathbf{T}\left(l_{k-1}, n_{p,k-1}, u_{q,k-1}, v_{r,k-1}\right) + \Delta t\right\} \tag{19}$$

where

$$\Delta t = \frac{\Delta l}{\left| \mathbf{t}_i \bullet \left\{ \frac{u_{q,k} + u_{q,k-1}}{2}, \frac{v_{r,k} + v_{r,k-1}}{2} \right\}^T \right|} \quad (20)$$

l, n and respectively denote the displacements along and perpendicular to $\mathbf{t}_i$, while u and v are the velocities along the x and y axes, respectively. k is the primary index. The algorithm evaluates the transitions from all possible combinations p, q, r of n, u and v at the $k-1$th step to all possible combinations of n, u and v at the kth step. It stores the minimum time for the admissible transitions in matrix $\mathbf{T}$ at each iteration k for use in the next iteration. The search is bi-directional (from both ends of the trajectory) in order to reduce the computational burden. When the searches from both directions coincide at the middle of the trajectory, a compatibility test is performed to determine if the searches have arrived at a feasible solution. If the test fails, the mesh is modified and the process is repeated. Once a feasible solution is obtained, a convergence check is performed. Convergence is determined by how close the current solution is to the trajectory obtained from the previous run of the optimization. If they are close enough, the solution is selected as the optimal trajectory. Otherwise, the mesh is refined around the new solution and the process is repeated until convergence is reached.

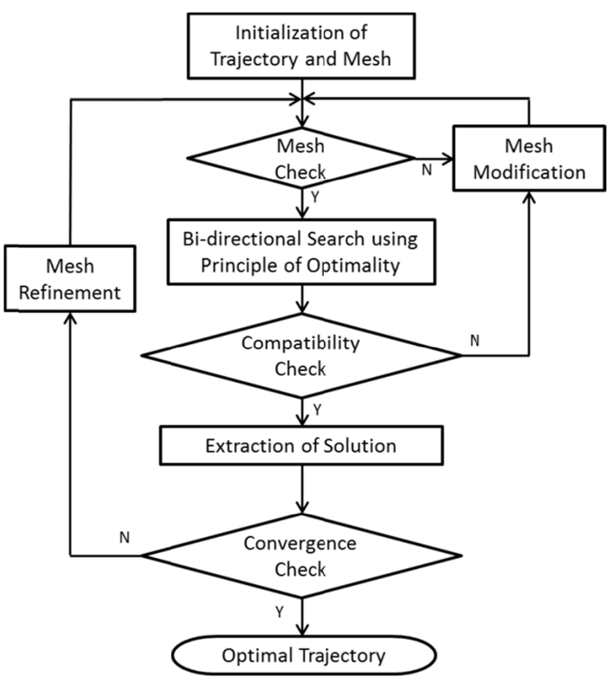

Figure 3. FLOWCHART OF THE DP ALGORITHM

5. RESULTS, CONCLUSIONS AND FUTURE WORK

A right-angled corner (with $\theta_1 = -90^\circ$, $\theta_2 = 0^\circ$) and an obtuse-angled corner (with $\theta_1 = 10^\circ$, $\theta_2 = 140^\circ$) both having $l_1 = l_2 = 0.1$ mm, $\varepsilon = 0.015$ mm, $V_{\max} = 25$ mm/s, and $A_x = A_y = 4000$ mm/s^2 are used to compare the following approaches:

(a) Optimal cornering using a pre-specified circular arc – the approach commonly used in industry
(b) Optimal cornering using the method proposed by Ernesto and Farouki [4] based on a pre-specified rational Bezier curve
(c) Optimal cornering using DP, as proposed in this paper.

The resulting corner paths and speed profiles for the right-angled corner are shown in Figs. 4 and 5. As can be seen, the proposed method based on dynamic programming has the shortest cycle time of all three optimal cornering approaches. It is 6.3 and 19.2% faster than the Bezier curve and circular arc methods, respectively. Note that a typical manufacturing process may have hundreds of thousands of sharp corners. Therefore, reductions in cycle time of even a few milliseconds per corner can quickly add up to significant savings in throughput. From their speed profiles, one observes that the Bezier curve is slowed down by its relatively tight curvatures while the circular arc is hindered by the long time it spends traveling at a fixed speed along the arc. The proposed approach is able to minimize cornering time by smartly modifying its shape autonomously. It is interesting to note that the path of the proposed method is similar to the time-optimal path followed by race car pilots when negotiating right-angled corners [8]. Even though it traverses a longer path, its curvatures are wider, allowing it to achieve higher speeds. Notice from Fig. 4 that even though the DP trajectory deviates from the two straight lines at points other than the critical point, the deviations fall within the tolerance constraint. This is because our choice of l_i = 0.1 mm is less than $l_{i,\max}$ = 0.157 mm. Figure 6 compares the overshoot in the y direction when traveling along the x axis from $\mathbf{p}_1$ to $\mathbf{p}_2$ for three values of l_i. As seen, $l_i \leq l_{i,\max}$ guarantees that the tolerance constraint is not violated and provides some safety margin because of the conservatism built into it.

The results from the obtuse-angled corner are shown in Figs. 7 and 8. V_c (calculated from Eq.(13)) is greater than $V_{\max}$ so l_i has no upper limit. This is because the DP optimization does not see much advantage in varying the curvature with regard to improving cornering time. As a result, it produces meager 2.2 and 2.8% improvements in cycle time over the circular arc and Bezier curves. Notice also that the velocity profile of DP trajectory in Fig. 8 is very coarse and that it violates the speed limit at a few instances. This is due to the poor resolution of the mesh used for the DP solution. The resolution of the DP solution can be improved, but this would only add to its already heavy computational burden. Note that the DP solution took about 10 minutes on a PC equipped with an Intel® Core™ i5-2400 processor @ 3.10 GHz and 8 GB RAM to generate the results shown in Fig. 7, compared to less than 1 second with the other two approaches. The high computational burden and poor resolution of DP are major limitations to using it for the purpose investigated in this paper. Future work will explore so-called indirect methods of optimal control (based on Pontryagin's Minimum Principle). Such methods hold the potential to achieve similar performance as DP with better resolution and far less computational cost.

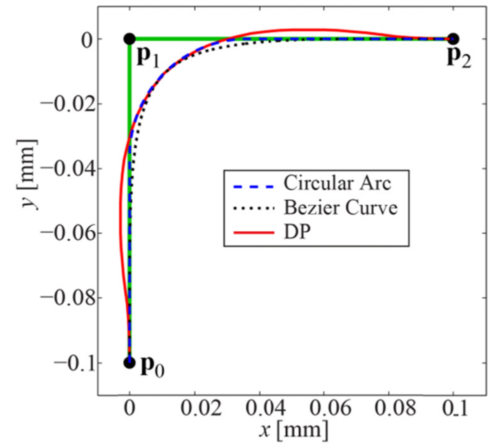

Figure 4. COMPARISON OF CORNER PATHS FOR RIGHT-ANGLED CORNER

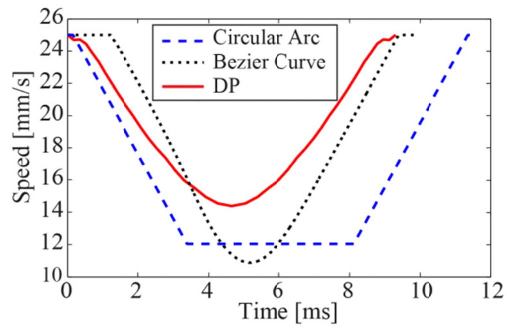

Figure 5. COMPARISON OF SPEED PROFILES FOR RIGHT-ANGLED CORNER

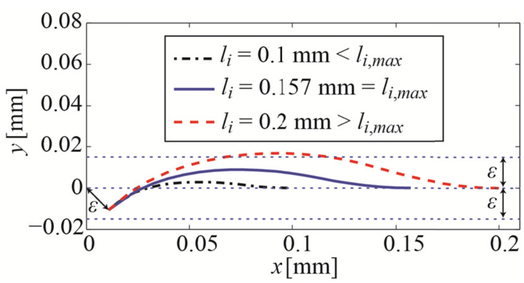

Figure 6. INFLUENCE OF l_i ON VALIDITY OF TOLERANCE CONTRAINT ASSUMPTION

Table 1. COMPARISON OF CORNERING TIMES

Corner	Cornering Time [ms]		
	Circular Arc	Bezier Curve	Dynamic Programming
Right-angled corner	11.51	9.92	9.30
Obtuse-angled corner	7.85	7.90	7.68

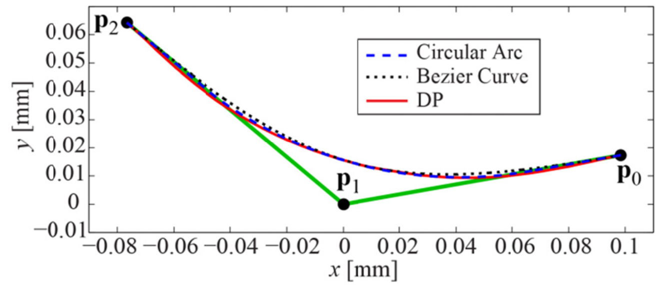

Figure 7. COMPARISON OF CORNER PATHS FOR OBTUSE-ANGLED CORNER

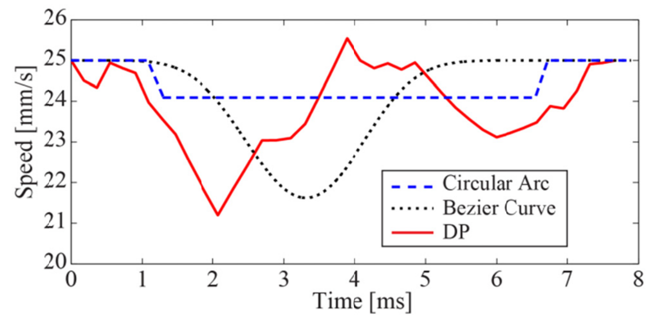

Figure 7. COMPARISON OF SPEED PROFILES FOR OBTUSE-ANGLED CORNER

6. REFERENCES

[1] Jouaneh, M. K., Wang, Z., and Dornfeld, D. A., 1990, "Trajectory planning for coordinated motion of a robot and a positioning table--I: Path specification," IEEE Trans. Robot. Autom., **6**(6), pp. 735–745.

[2] Erkorkmaz, K., Yeung, C. H., and Altintas, Y., 2006, "Virtual CNC system. Part II. High speed contouring application," Int. J. Mach. Tools and Manuf., **46**(10), pp. 1124-1138.

[3] Imani, B. M., and Jahanpour, J., 2008, "High-speed contouring enhanced with PH curves," Int. J. Adv. Manuf. Technol., **37**(7-8), pp. 747-759.

[4] Ernesto, C., and Farouki, R., 2012, "High-speed cornering by CNC machines under prescribed bounds on axis accelerations and toolpath contour error," Int. J. Adv. Manuf. Technol., **58**(1-4), pp. 327–338.

[5] Bobrow, J., Dubowsky, S., and Gibson, J., 1985, "Time-optimal control of robotic manipulators along specified paths," Int. J. Rob. Res., **4**(3), pp. 3-17.

[6] Dong, J., and Stori, J., 2006, "A generalized time-optimal bidirectional scan algorithm for constrained feed-rate optimization," J. Dyn. Syst. Meas. Control, **128**(2), p. 379

[7] Bellman, R., and Dreyfus, S., 1962, "Applied dynamic programming.

[8] Barre, P. J., Dieulot, J. Y., Bearee, R., and Bouzidi, A., 2005, "A heuristic path-planning method for enhancing machine-tool contour-following," J. Syst. Sci. and Syst. Eng., **14**(1), pp. 85-96.