The Synthesis of a Standard Trajectory Used in SCARA Systems

Jean-François Gauthier†, Jorge Angeles†, Scott Nokleby‡*
†Department of Mechanical Engineering and Center for Intelligent Machines, Mcgill University
‡Faculty of Engineering and Applied Science, University of Ontario Institute of Technology

1 Introduction

The test trajectory for a SCARA system involves a vertical upward translation of 25mm, a horizontal translation of 300mm and a final vertical downward translation identical to the first one. The system has to move through this symmetric trajectory back and forth with a rotation of the end-effector of 180° in 500ms with a payload of 2kg. This test trajectory includes square corners between the vertical and horizontal segments, which are obviously sources of acceleration discontinuities. These corners have to be smoothed in order to provide *Second-order geometric continuity* G^2 [1] throughout the test trajectory. To simplify the equations in the later sections, as shown in Fig.1, we will work with the first half of a smoothed trajectory by using the symmetry plane of the trajectory along the z axis. This means

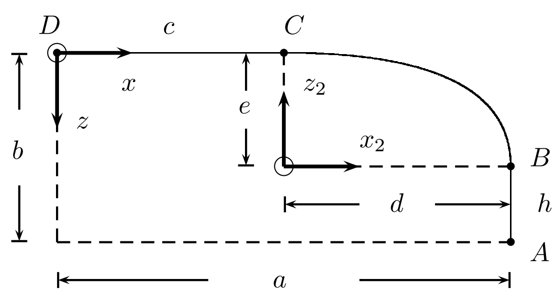

Figure 1: Half of the proposed Trajectory with its parameters

that the parameters value in Fig. 1 are $a = 150$mm and $b = 25$mm. Also, the time t_{AD}, which is the time required to do half of the trajectory, obviously passing through points A-B-C-D, is fixed and is equal to a quarter of the cycle time of the test trajectory, which means that $t_{AD} = 125$ms. Finally, the rotation of the end-effector between points A and D will be 90°.

2 Trajectory Smoothed with Lamé Curves

Lamé curves are defined by the equation

$$u^m + v^m = 1 \qquad m = 1, 2, \ldots \tag{1}$$

As m increases to infinity, equation (1) leads to a square shape. In our case, a value of $m = 3$ will be used because it provides G^2-continuity [1] without increasing too much the complexity of the equations. For given parameters $d = a - c$ and $e = b - h$, shown in Fig. 1, the equation that we will use to smooth the curve in the coordinate frame x_2-z_2 is

$$\left(\frac{x_2}{d}\right)^3 + \left(\frac{z_2}{e}\right)^3 = 1 \tag{2}$$

*E-mail adresses: jfgauthier@cim.mcgill.ca, angeles@cim.mcgill.ca, scott.nokleby@uoit.ca

For equation (2), we can find x_2 and z_2 in terms of θ [2] by applying an affine transformation scaling every value of x_2 and z_2 correspondingly to parameters d and e, such that

$$x_2(\theta) = \frac{d}{(1+\tan\theta^m)^{1/m}} \tag{3a}$$

$$z_2(\theta) = \frac{e \cdot \tan\theta}{(1+\tan\theta^m)^{1/m}} \tag{3b}$$

These relations are very important and will be used in the next section.

3 Position, Velocity and Acceleration w.r.t. the Trajectory

To be able to associate a cartesian position, velocity and acceleration to any given point on the trajectory based on a given velocity profile defined over the trajectory, we need to know the relation between the position, velocity and acceleration of the Lamé curve w.r.t. the nominal displacement $s(t)$, velocity $\dot{s}(t)$ and acceleration $\ddot{s}(t)$ of the velocity profile over the constrained time.

Since the cartesian position equations of x_2 and z_2 in terms of θ have already been defined with equations (3a) and (3b), what we need here is a relation between θ and s such that $\theta = \theta(s)$. Of course, such relation is usually impossible to find mathematically. Our approach here uses the fact that we work incrementally in time, which gives us a finite number of points over the trajectory. Then, we can converge numerically with the Newton-Raphson optimization method at each point of the trajectory that lies on the Lamé curve with the use of equations (3a) and (3b).

Assuming that $s(t_k)$, $\dot{s}(t_k)$ and $\ddot{s}(t_k)$ are given for every $k = 1, 2, \ldots, N_{AD}$, where N_{AD} is the total number of points over the trajectory. We need to consider that the Lamé curve is between point B and C of the trajectory, as we saw in Fig. 1. If we state that all the time-increments $\Delta t = t_{k+1} - t_k$ are equal and assume that the values of t_{AB} and t_{AC} are known, then, we can define two points N_1 and N_2 associated to two distinct time limits t_{N_1} and t_{N_2} such that

$$t_{AB} \leq t_{N_1} < t_{AB} + \Delta t \tag{4a}$$

$$t_{ABC} - \Delta t < t_{N_2} \leq t_{ABC} \tag{4b}$$

The two points N_1 and N_2 will actually be the limits of the algorithm using the Newton-Raphson optimization method to find all the angles θ_k associated to each displacements s_k such that

$$0 \leq \theta_k \leq \pi/2$$

with

$$s_k = s(t_k) - s_{AB} = s(t_k) - h \qquad k = N_1, \ldots, N_2 \tag{5}$$

and where s_{AB} defined as $s_{AB} = s(t_{AB}) = h \neq 0$. This relates the angle θ_k to the displacement made on the trajectory after point B. We can also associate the length of the curve, which is, as shown in Fig. 2, the same as the displacement on the trajectory from point B, to the following equation

$$s_k = \int_0^{\theta_k} \sqrt{\left(\frac{\partial x_2(\theta)}{\partial\theta}\right)^2 + \left(\frac{\partial z_2(\theta)}{\partial\theta}\right)^2} d\theta \tag{6}$$

In equation (6), the only unknown is θ_k. Obviously, solving this equation analytically for θ_k is certainly very difficult or maybe even impossible. This is where we introduce the Newton-Raphson optimization method to numerically find the value of θ_k for every $k = N_1, \ldots, N_2$.

First, we need to define our objective function. To simplify the presentation, we will use $\vartheta = \theta_k$, where ϑ will now be the unknown we will be searching for. If we also modify equation (6) to make it equal to zero, we obtain our objective function

$$f(\vartheta) = \int_0^{\vartheta} \sqrt{x_2'(\theta)^2 + z_2'(\theta)^2} d\theta - s_k = 0 \tag{7}$$

Basically, the Newton-Raphson method is an iterative method that approaches the solution of determined nonlinear systems in a finite number of iterations. In our case, we have one unknown, ϑ, and one equation, equation (7). At every iteration, the value of ϑ is updated such that

$$\vartheta_{i+1} = \vartheta_i - \frac{f(\vartheta_i)}{f'(\vartheta_i)} \tag{8}$$

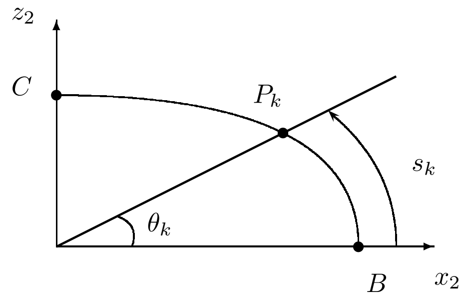

Figure 2: θ_k versus s_k

where

$$f'(\vartheta) = \sqrt{x_2'(\vartheta)^2 + z_2'(\vartheta)^2} \tag{9}$$

Equation (8) is repeated iteratively until

$$\left| \frac{\int_0^{\vartheta_i} \sqrt{x_2'(\theta)^2 + z_2'(\theta)^2} d\theta - s_k}{\sqrt{x_2'(\vartheta_i)^2 + z_2'(\vartheta_i)^2}} \right| \leq \epsilon \tag{10}$$

where ϵ is the tolerance desired to stop the algorithm.

Now, since we know the relation between θ_k and $s(t_k)$, when $k = N_1, \ldots, N_2$, we can now figure out the cartesian position of every time incremented points of the trajectory with respect to the x-z coordinate frame shown in Fig. 1, assuming there is a given velocity profile. With respect to time, from point A to point D, the equations for $x(t)$ and $z(t)$ can be defined as

$$x(t) = \begin{cases} a & \text{if } t \in [0, t_{AB}] \\ a - (d - x_2(\theta)) & \text{if } t \in (t_{AB}, t_{AC}] \\ c\left(1 - \frac{t - t_{AC}}{t_{CD}}\right) & \text{if } t \in (t_{AC}, t_{AD}] \end{cases} \tag{11a}$$

$$z(t) = \begin{cases} b - s(t) & \text{if } t \in [0, t_{AB}] \\ b - (h + z_2(\theta)) & \text{if } t \in (t_{AB}, t_{AC}] \\ 0 & \text{if } t \in (t_{AC}, t_{AD}] \end{cases} \tag{11b}$$

4 The Next Step

Future work will consist in the optimization of the proposed trajectory w.r.t. the lengths of the horizontal and vertical segments c and h. Different velocity profiles will be used to determine the set of optimized parameters that minimizes the rms value of the time-rate of change of the kinetic energy over the whole trajectory.

References

[1] C.-P. Teng, S. Bai, and J. Angeles. Shape synthesis in mechanical design. *Acta Polytechnica*, 2006.

[2] W.A. Khan, S. Caro, D. Pasini, and J. Angeles. Complexity analysis of curves and surfaces: application to the geometric complexity of lower kinematic pairs. *Department of Mechanical Engineering and Centre for Intelligent Machines Technical Report*, 2006.