Contents lists available at ScienceDirect

Journal of Manufacturing Systems

journal homepage: www.elsevier.com/locate/jmansys

Technical Paper

Exponential and sigmoid-interpolated machining trajectories

Christopher DiMarco[a,b,*], John C. Ziegert[a,b], Christopher Vermillion[a]

[a] The University of North Carolina at Charlotte, Charlotte, NC, USA
[b] Center for Precision Metrology, USA

ARTICLE INFO

Article history:
Received 25 March 2015
Received in revised form 25 March 2015
Accepted 25 March 2015
Available online 7 August 2015

Keywords:
Machining
Trajectory planning
Tool path generation
Interpolation techniques

ABSTRACT

In single-point metal turning and boring processes, a chip nest can often be created that is a hazard to part and operators alike. In order to mitigate this, a process called modulated tool path (MTP) machining was developed that superimposes a sinusoidal motion tangent to the feed direction onto the tool feed path to break chips. The sinusoidal motions are created under CNC control in the part program. In the current implementation, the sinusoidal motion is approximated as a series of short linear moves. Linear interpolation is currently used to create position and velocity commands to the axis servomotors at each control loop closure. Linear interpolation is a computationally heavy and dated method that is not well tailored to a sinusoidal trajectory. In this paper a new method called the sigmoidal interpolator is introduced that honors all physical constraints of a machining system while offering better tracking performance and lower accelerations than the linear interpolator, all while reducing the number of possible state transitions of the implemented software from approximately 17 to 4.

1. Introduction

Metal turning and boring processes can often lead to a 'chip nest,' or a long strand of metal that can wrap around the tool tip or part. A chip nest can damage the part finish and also presents a safety hazard, as machine operators removing the nest can be injured by the razor-sharp strand (see Fig. 1). One solution to this issue is called modulated tool path (MTP) machining, a process by which the tool tip is oscillated in the direction of the instantaneous tool feed motion. A new chip is formed each time the tool enters and exits the cut [9].

In a single-axis tool path where a normal cutting motion is a position vs. time ramp, the oscillation is added by superimposing a sinusoid on the ramp. The frequency and amplitude of the oscillation, as well as the spindle speed, control the length of the chip that is removed and can also have an impact on the surface finish of the part [2].

The NC part program generally only provides the start and end coordinates of linear moves along with the desired feed rate during that move. The servo-control system requires a position and velocity to be specified at each closure of the control loop. These intermediate points and velocities are generated by an interpolation algorithm built into the controller. There are many interpolator implementations. When smooth and vibration-free motions are desired, it is common to use an algorithm that places limits on allowable jerk, while simultaneously respecting the physical acceleration and velocity capabilities of the machine. This algorithm is outlined by Altintas [1]. When it is desired to maximize fidelity to the nominally commanded path, the jerk curve is rectangular with only three values for jerk, j(t): ±jlim, or zero. This in turn leads to a trapezoidal-shaped acceleration profile with no discontinuities that would require instantaneous changes in the force or torque. The velocity and position profiles take on parabolic and cubic characteristics respectively. A maximum acceleration may also be instituted, but often it is directly evaluated from the maximum allowable jerk [1]. A sample single-axis trajectory is shown in Fig. 2.

The linear interpolator is the subject of a significant body of literature. Researchers seeking to improve machine positioning performance have investigated advanced controller design [4], improving contour accuracy [5,7], input shaping [6], and high performance [3]. While the linear interpolator is conceptually simple, practical implementation relies on a large network of if/then statements to determine at what times during each move to switch from one jerk value to another in order to assure the axis reaches the desired future positions and velocities without overshoot. This requires approximately 17 different branching conditions for all possible states of position, velocity, acceleration, and jerk. Since

* Corresponding author at: The University of North Carolina at Charlotte, Charlotte, NC, USA.
E-mail addresses: cdimarc1@uncc.edu (C. DiMarco), jziegert@uncc.edu (J.C. Ziegert), cvermill@uncc.edu (C. Vermillion).

http://dx.doi.org/10.1016/j.jmsy.2015.04.007
0278-6125/© 2015 The Society of Manufacturing Engineers. Published by Elsevier Ltd. All rights reserved.

Fig. 1. A chip nest forming on a machine tool during a turning operation. After a few moments of cutting, these razor sharp strands become tangled with the tool and workpiece.

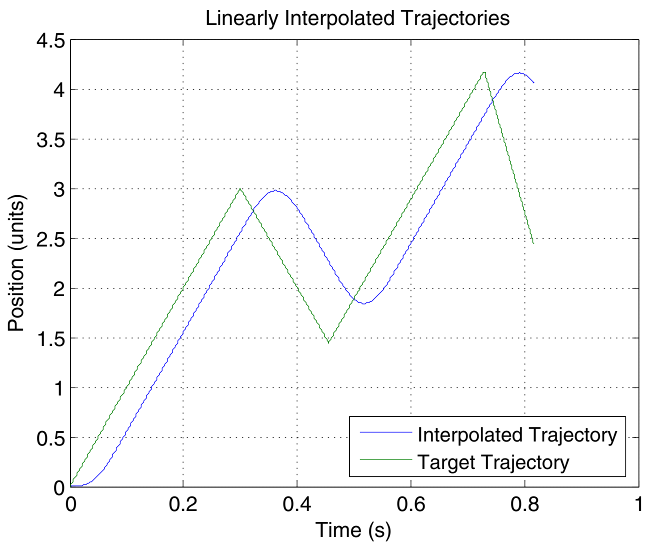

Fig. 2. A sample linear interpolated movement. The green line is the desired movement as provided by the NC program. The blue curve is the jerk-limited linear interpolator output. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of the article.)

branch conditions are the most expensive processes in terms of computing time, this invariably leads to large computation time costs. Secondly, there is often an interpolator delay created by the acceleration and jerk limitations that is never recouped. In Fig. 2, this is readily seen by the growing time differential between the desired curve (green) and the interpolated curve (blue). Typical linear interpolator implementations accept that the idealized motions embodied in the part program are physically unrealizable because the instantaneous velocity changes would require infinite acceleration, and make no attempt to bring the actual tool trajectory back onto the idealized motion when away from the transients at the beginning and end of each segment.

This paper reviews strategies for jerk and acceleration limited trajectory planning called the exponential interpolator developed by Rymansaib [8], and proposes a sigmoidal interpolator to replace the planning previously done by linear interpolators at transitions. This technique will be shown to be more appropriate for periodic inputs, as well as being useful for standard machining processes, and requiring fewer branching conditions.

2. Sigmoidal interpolators

Conceptually, the proposed interpolator works by multiplying the nominal motions described in the NC part program by a sigmoidal, or S-shaped, “blending” function that smoothly transitions from zero to one, one to zero, or zero to one and back to zero. The result is a continuously differentiable curve that obeys the initial conditions of the commanded motion and reaches the desired nominal trajectory when the value of the function reaches one. Many sigmoid-type functions exist, including the logistic function, the arctangent, the hyperbolic tangent, and the error function. Slightly different implementations are used for the case of startup from rest, and transition from one commanded velocity to another.

2.1. Startup

For modulated tool path machining, each motion segment is composed of a linear component for nominal feed and a superimposed sinusoidal component for chip breaking. Making use of superposition, these two components can be evaluated separately and then summed to get position, velocity, acceleration, and jerk values.

Linear component. The linear component of the exponential interpolator is detailed here, as well as by Rymansaib [8]. The baseline path for the interpolator to match is given by the linear equation in (1), where x is the resulting position and m is the linear slope:

$$x = mt + b \tag{1}$$

For initial startup, the equation for the sigmoid interpolator, SL is:

$$S_L = 1 - e^{-\alpha t^3} \tag{2}$$

where α is a time constant, and t is the time. The order of the time variable t is chosen as 3 to maintain a zero value at $t = 0$ for acceleration, as will be shown below. This leads to a sigmoid interpolated position value x_L of:

$$x_L = xS_L = (mt + b)(1 - e^{-\alpha t^3}) \tag{3}$$

Note that at $t = 0$ and for sufficiently large values of t, $x_L = x$. The velocity, $\dot{x}_L$, is:

$$\dot{x}_L = m - e^{(-\alpha t^3)}(m - 3\alpha m t^3 - 3\alpha b t^2) \tag{4}$$

The velocity is equal to 0 at $t = 0$, and asymptotically approaches the commanded feedrate, m. The value of α determines how quickly this occurs. Differentiating a second time, the acceleration, $\ddot{x}_L$, is equal to:

$$\ddot{x}_L = e^{-\alpha t^3}(-9\alpha^2 b t^4 - 9\alpha^2 m t^5 + 6\alpha b t + 12\alpha m t^2) \tag{5}$$

Again, at $t = 0$, the acceleration is equal to zero. With $\ddot{x}_L$ limited to a physically realizable maximum acceleration and the values of t known for the movement, the maximum allowable value for α can be determined using a numeric solver.

The linear paths generated by the interpolator are shown in Fig. 3. Note that the sigmoidal interpolator causes the feedrate of the tool to exceed the desired value for some period of time and it tries to “catch up” to the commanded trajectory. As α increases, the planned trajectory more quickly reaches the desired path.

Sinusoidal component. The desired sinusoidal motion is given by:

$$x = A\ \sin(\omega t) \tag{6}$$

where A is the sine wave amplitude and ω is the angular frequency. As with the linear component, the sinusoidal component is multiplied by a sigmoidal function to control velocity and acceleration. The form is similar:

$$S_S = 1 - e^{-\alpha t^2} \tag{7}$$

However, the order of t is decreased to 2 as it is not required to be increased to maintain zero acceleration at time step zero once