Spline Functions Generate Smooth Moves for Motion Controllers

By Peter Nachtwey, Delta Computer Systems, Inc.

Machine builders need to increase the performance and precision of motion in their systems in order to maximize processing throughput while minimizing waste. To accomplish this – and make systems easy to design and tune – some motion control manufacturers are incorporating spline functions into their control algorithms.

Spline functions emulate flexible strips of wood (splines) that can be bent between fixed points and used to draw smooth curves. With spline functions, implementing smoothly curving motion profiles is as easy as providing the motion controller with position coordinates (as a function of time or another axis' position) and instructing it to connect the dots. The motion controller is simpler to program since the system designer doesn't need to calculate multiple velocities and acceleration rates. And system performance improves because a single spline function replaces many point-to-point motion steps.

The motion controller uses the spline to do cubic interpolations between the defining points as a function of time or another axis' position. Velocities and accelerations are determined by differentiating the spline equation at each interpolated position. Complex motion profiles can be easily specified graphically with a CAD tool or with an Excel spreadsheet. The machine designer defines the positions and lets the spline algorithm compute the acceleration and velocity necessary to get smoothly from one point to another. As the spline function executes, the motion controller calculates the acceleration and velocity needed to move from the current point to the next point in the motion profile, such that the accelerations are continually varying at a linear rate for smooth motion curves. (See spline plot below).

The difficulty of calculating splines depends on the spline algorithm that is used and whether the CPU doing the calculations has floating point capabilities. The basic algorithm for cubic splines is found in 'Numerical Recipes in C' by Press Teukolosky Vetterling and Flannery. It is necessary to cube some numbers, but this is still within the capability of most 16 bit micros or DSPs; higher order splines require floating point math. Using cubic splines first requires calculating the acceleration at each point, then doing a cubic interpolation between two of the defining points. These calculations are relatively fast. The motion profile may not be accurate outside the current point interval, and the calculation time of the accelerations at each point is proportional to the number of points defining the spline.

Splines can be used to "linearize" an application such as a pouring molten metal from a crucible. The pouring motion needs to be non-linear in order to maintain a uniform flow. The trick is to use two axes: a virtual master axis with no hardware attached which provides a position to which the second axis, the slave axis, is electronically "geared." The spline curve is used like a look-up table to convert rotary positions to linear positions so commands can be issued to pour as a function of angle instead of linear position. The

slave axis uses the spline and the virtual master's angle to index into the spline curve to find the real linear position to which that the axis must move.

Another application is curve sawing. The yield of finished lumber from a curved log can be maximized if the wood is cut to the desired width and thickness following the grain or curve of the log and then dried straight. Typically, an optimizing computer scans the log to determine the best way to cut the lumber then downloads spline parameters to the motion controller that smoothly moves chipper and saw blades to cut the log along smooth curves, geared to the chain that is carrying the lumber. If the speed of the feed system changes, all the axes executing splines as a function of the feed system will also change proportionately.

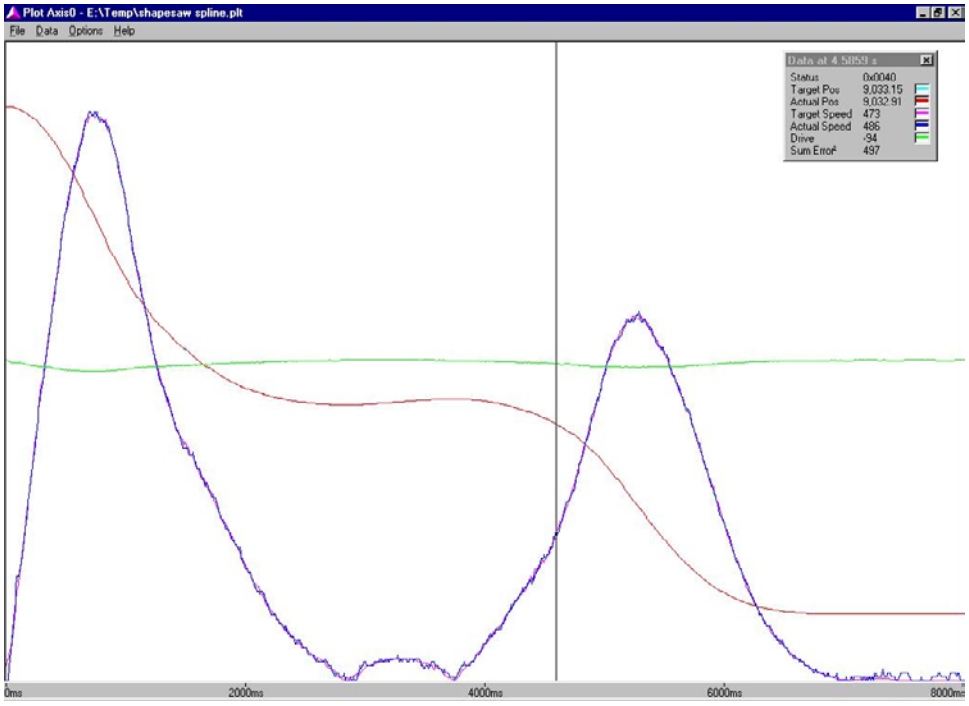

Fig. 1 Plot of Spline Curve from Curve Sawing Application