Software Library Description

The group of Algorithms (software routines) structures that follow, show how to perform "on-the-fly" math using an eight bit 12 Mhz processor. The reason for the inclusion of these "algorithms" is to help you derive methods to ...

1) Reduce hardware cost (elimination of a math co-processor)
2) Acquaint you with binary 2s Logarithm math
3) Show you a method for deriving "hi-speed" math routines

These routines were written for servo applications utilizing the National LM628/29 Servo Control Processors. They show the coordination required for four interpolated moves .... Circular Interpolation, Cubic Spline Interpolation, HandWheel (sometimes referred to as "Electronic Gearing"), and Registration. A brief explanation of these move styles follows below. The actual codes given are, in some cases, shown in "C", and Basic, but are always described in Intel 8051 assembly language instruction code.

****

I) HandWheel (Master-Slave, or Electronic Gearing)

HandWheel operation describes the tracking (or mimicking) of one axis of motion by another. The tracking (or slave) axis can be programmed to follow the motion of the master axis in either a velocity, or position mode as required. Another programmable entry allows the slave axis to follow the master in the same, or opposite direction of rotation, while one more entry allows the tracking ratio between the master, and slave axis to range from 1:65536 to 65536:1.

HandWheel operation is useful in systems requiring minimal control yet precise coordination of more than one independent motor motion, such as multiple product line synchronization (packaging).

II) Registration

Registration is a term used to describe the maintaining of position between two independent products. An example of the registration technique is as follows. Place two pencils, tip to tip, on a table. Have someone move one of the pencils across the table, while you, at the same time, try to maintain the exact position relationship that the pencils had at the start of the move.

The Registration algorithm allows user entries for tolerance, percentage of recovery, maximum error for no-recovery abort, and offset allowance between the two products being registered. Registration is used where the exact positional relationship of two or more products must be maintained (such as packaging, or the manufacture of bandage products).

III) **Circular Interpolation**

Circular interpolation describes the coordination of two independent motor axis to produce an apparent circular motion. An example of how circular interpolation works, is to hold a pencil in a fixed vertical position, while moving a piece of paper (placed under the pencil tip) to draw a circle. Moving the paper left and right describes one axis, while moving the paper toward you and away from you constitutes the other.

The Algorithm will perform the circular move at a specified velocity, with arc center-point and ending-point entries describing the specific arc itself (full, or partial circle motion). Another entry sets the increment to allow the user to set the angular step with which the calculations will be done (all of the circular interpolation math is accomplished **"on-the-fly"**). The only angle restriction, therefore, is the resolution of the feedback device, and the system mechanics.

IV) **Cubic Spline**

The Cubic Spline algorithm, allows the controller to form a continuous (curved) path interpolation between sets of designated point positions. Sometimes referred to as curve fitting, the Cubic Spline routine allows the digitizing of an arbitrary path with a minimum of position point entries.

A table of fixed points through which the developed curve must pass, the number of points **between** these fixed points that the computer must calculate and also go through, and the specific velocity at which to do the curve are the programming entries required.

V) **Logarithm Math**

Utilizing Logarithms to solve multiple byte multiplication and division will significantly enhance the real time capability of your operating software. When doing 4 byte math, a worst case multiply routine can be sped up by a factor of 8, and a worst case divide by a factor of 14 over the standard shift and add/subtract bit manipulation method. Using an 8052 series of processor (with a 12 Mhz clock) a number to log (or log to number) convert takes about 40 microseconds (us). The intermediary adds and subtracts (multiply and divides) take about 15 us. A single mult. or div. operation can then be as low as 135 us, a square (or sqr root) function about 100 us, and a cube (or cube root) 115 us.