We have had quite a lot of discussion on point-to-point (PTP) CNC machines. However, the versatility of the CNC machine is fully realized once we start talking about continuously controlled machines. What are continuously controlled machines ? These are the ones in which you are not only controlling the initial and final positions of the tool/cutter but also the path along which you are moving it. And *also the velocity of the tool along this particular path.*

But controlling the end position of the tool - isn't that good enough ? Not always. I mean – alls well that ends well – must have been uttered by a PTP machinist. And the continuous control programmer must have retorted - The end does not always justify the means.

Jokes aside, let us take the case of milling of profiles, for example. If you are cutting all around a metal plate and getting a *particular* shape out of it, it is necessary that you control the path of the tool continuously from the initial point to the final one. But that has been done for decades on conventionally controlled milling machines – so why are we so much bothered to control the path of the tool in case of CNC machines ? This is because the milling machines of yester years (that is, milling machines without CNC but with automatic control of some sort or simply manual control) had their motions controlled by some sort of physical devices or means while we are now trying out the same thing by digital control in case of CNC.

Basically – in CNC - there are three motors connected to three axes X, Y and Z and we provide appropriate signals to these three motors so that their combined velocity is exactly equal to the programmed velocity. And their ratios are such as to guide the tool along the programmed path.

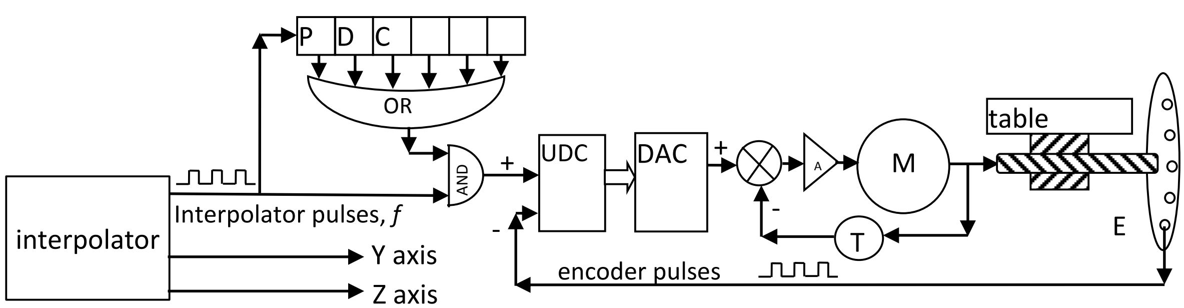
In order to do this – we need a device which can provide signals to the X, Y and Z axis motors with correct ratio and with correct signal values so as to produce the programmed velocity value of the cutter path along the programmed path.

What does all this mean ? It means that we are going to send some signal to the motors to make them rotate. Even if these motors are DC motors or AC motors, we can simply run them by sending a train of voltage pulses. Please see the fig.1 and you will agree and understand.

Let us consider a couple of CNC program instructions

N001 G90 G01 X20 Y40 F100
N002 X80 Y80

Fig. 1 The X axis control loop and interpolator



This figure is for the X axis control loop only. From the interpolator, some pulses are coming in. These pulses are being checked for their total number by the Position down counter (PDC) and then these pulses are continuing to the UP-counting input of the UP-DOWN counter (UDC). This circuit now processes this pulse rate and number of such pulses to properly rotate the X axis motor at its required speed for a particular extent of rotation.

At first, the interpolator pulses come in and fill up the Up-Dn counter to a high content which converted to a high voltage by the DAC (digital to analog converter). This voltage drives the motor and ultimately makes the table move and the encoder rotate. The rotating encoder sends back pulses and these pulses reduce the content of the Up-Dn counter.

When the interpolator pulse rate and the encoder pulse rates are same, the axis settles down to a steady voltage output of the Up-Dn counter – DAC duo and the motor executes a steady angular speed resulting in constant speed movement of the table.

Q1. There is a control loop along X axis with PMDC motor connected to single start lead screw (pitch is p = 4 mm) via gearbox of ratio 1/3 and with encoder mounted on leadscrew with h=200 holes on the circum. Find out the interpolator pulse frequency to the X loop for the command

N0001 G90 G00 X20 Y300
N0002 G01 X50 Y340 F200

This means that distance along X to be covered is 30 mm while it is 40 mm along Y.
It is linear Interpol. Feed along cutter path is 200 mm/min. Total dist = $\sqrt{(30^2+40^2)}$=50 mm

As the displacement triangle and velocity triangle are similar in case of linear Interpol

Vx/V = dx/d OR Vx = V*(dx/d) = 200*(30/50)=120 mm/min

Hence, when the table moves along X @ 120 mm/min, the lead screw rotn is 120/4 = 30 rpm

Hence – in 30 rpm, the encoder would send 30 X 200 = 6000 pulses per minute

At steady state, encoder pulse rate = interpolator pulse rate

Therefore, Vx would mean interpolator sends to the X loop → 6000 ppm

There are similar loops for the Y and the Z axes with pulse rates coming in from the interpolator. These pulse rates are different for the different axes, according to the particular segment of profile being cut.

In the present case (Fig. 1, not the numerical problem), since the X motor and Y motor have to move the table by 60 mm and 40 mm respectively, the velocity triangle and displacement triangles are similar, hence the ratio of the speeds of X and Y motors is 3:2. Hence the ratio of the pulse frequencies sent out of the interpolator for the X and Y control loops will have to be 3:2. Here we are assuming that all the axes drives are identical.

Now for the basic way in which this interpolator works:

Interpolator has to send out trains of pulses to the control loops of the three axes of motion.
The interpolator itself is provided, as input, a fixed high frequency clock pulse rate for operation. All the operations of the interpolator are to be carried out to the tune of this clock pulse rate.

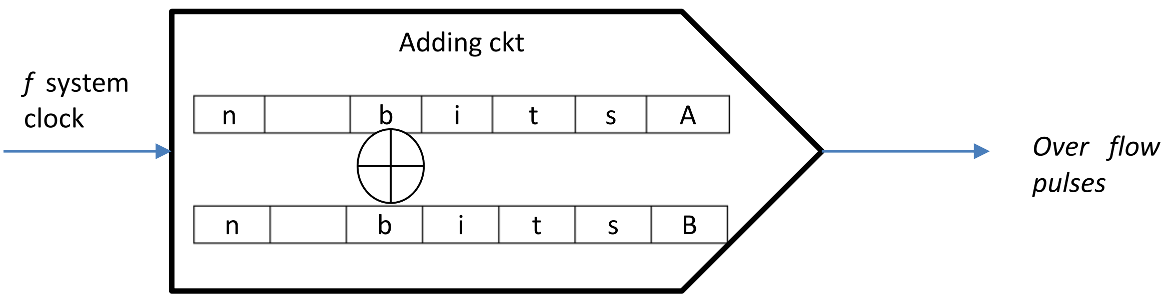
The interpolator works this way : if you want a high feed rate along a particular axis of motion, a proportionally high-frequency pulse train is to be sent from the interpolator to that axis control loop.

Obviously, the frequency of this second pulse train (which goes from the interpolator to the control loop) has to be less ~~or equal to that of~~ the system clock at the input of the interpolator.

Hence, depending on the requirement, the interpolator has to send out a pulse train of frequency anywhere between 0 and system clock frequency.

How do we reduce frequency of a train of pulses ? One simple way would be to make the mother train of pulses carry out some bogus repetitive operation of fixed time interval and come out of the task whenever some goal is reached.

So the basic shape of the interpolator unit cell emerges as an adder



With each pulse of the clock, one addition of the numbers in counters A and B takes place and is dumped into counter A.

At the start of every line, the machine puts in some number inside counter B and sets counter A to zero. Additions take place and ultimately counter A overlfows. This overflow pulse is connected to the output and one pulse is outputted.

As the size of the counters is n bits, they have capacity of 2^n-1. If you try to put in something more on top of that – they would overflow.

Hence,

When 1 is put into counter B and added again and again, we would require at least 2^n additions to create 1 overflow

When p is put into counter B and added again and again, we would require at least $2^n/p$ additions to create 1 overflow

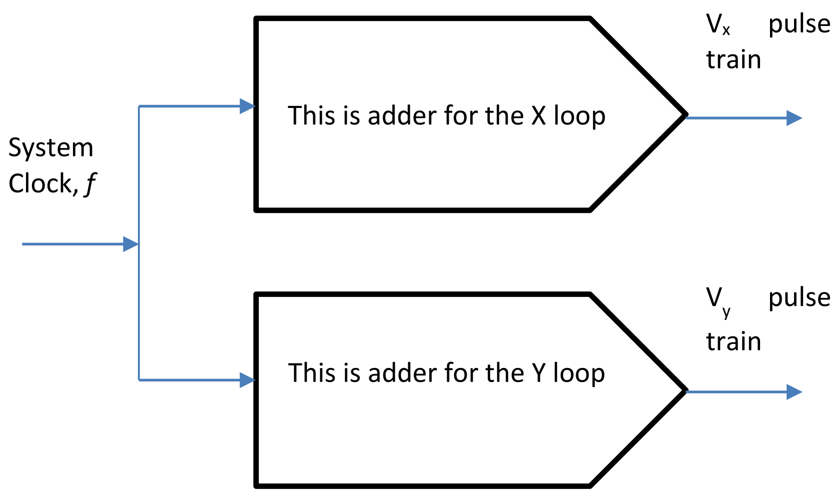
Hence, for $2^n/p$ clock pulses – 1 overflow is produced

Hence, for 1 clock pulse – $p/2^n$ overflow is produced

Hence, for f clock pulses – $f \times p/2^n$ overflow is produced

If that be so

Lets have the basic interpolator for X-Y table in the following form



In that case – we may write that the Vx pulse train is nothing but

$$V_x = \frac{f \times px}{2^n}$$

Where px is the content of the counter B of adder for the X loop

What is the value of px ? I don't know – you have to find out

In order that the cutter travels along the correct path – the velocity values along X and Y axes need to have the correct ratio. This ratio Vx/Vy is nothing but Δx/Δy for linear interpolation.

Hence, put px = Δx and py=Δy and the ratio between Vx and Vy pulse rate values would be as per programmed value.

But Is that all ? Not at all. The absolute value of V along the path has to be the programmed feed value. How much is that ? Whatever is programmed.

Say V = 200 mm/min is programmed. It would have a corresponding pulses / min value.

We should have $V_x^2 + V_y^2 = V^2$

So, $\left\{\frac{f \times \Delta x}{2^n}\right\}^2 + \left\{\frac{f \times \Delta y}{2^n}\right\}^2 = V^2$

However, this poses a problem. We have a relation

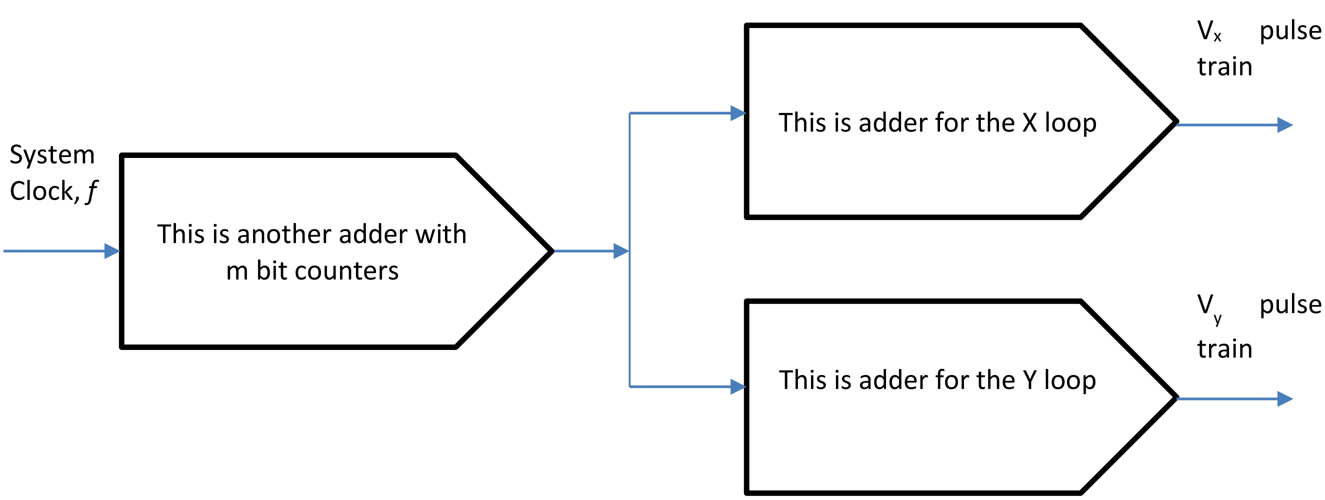
$$\frac{f}{2^n} = \frac{V}{\Delta L} \dots \dots \dots (1)$$

Where , $\sqrt{\{\Delta x\}^2 + \{\Delta y\}^2} = \Delta L$

In eqn (1), we have the left hand side a constant, where f is set once the manufacturer selects it. It is the system clock frequency. The power of 2 in the denominator depends on hard-wired circuitry – the number of bits in the counters of the adding circuits of interpolators. Hence, the ratio V/ΔL is being equated to a constant. But V/ΔL cannot be a constant from one program line to another. How can you have the velocity and segment length of cut to be always in a definite ratio. That's not just possible.

This points to the fact that the circuit we have considered for the interpolator does not have adequate flexibility. So why not add another variable to the whole set of relations.

Have a look at this



There !

One more adder is added and that adder accepts a number called F.R.N (feed rate number) which it adds again and again to itself till 2^m is reached and overflow pulse is sent to the X and Y adders. These adding circuits are called digital differential analyzers in books by Y Koren.

So the incorporation of this digital circuit element results in

$$\left\{\frac{f \times FRN \times \Delta x}{2^{m+n}}\right\}^2 + \left\{\frac{f \times FRN \times \Delta y}{2^{m+n}}\right\}^2 = V^2$$

So that

$$\frac{f}{2^{m+n}} FRN = \frac{V}{\Delta L} \dots \dots \dots (2)$$

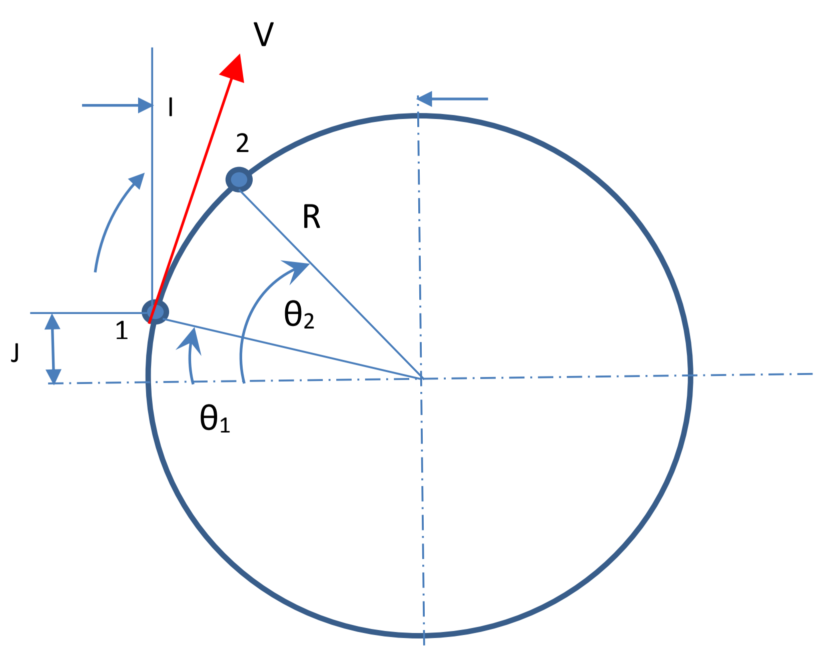
Now the previous cause of tension is not there anymore. Why not pile the constant terms together and set them to a particular constant value ? Hence, let

$$\frac{f\,(ppm)}{2^{m+n}} = \frac{1}{10}$$

This gives us

$$FRN = \frac{10V}{\Delta L}$$

So this means that in case of any line segment, just define the FRN as above, define the adding numbers of X and Y DDAs as Δx *and* Δy respectively and your system will work. In the previous



case, it wouldn't have worked (eqn (1)) as it imposed a constraint on the relation between V and ΔL which was not practicable.

If you are intending to cut a circular profile, say in the third quadrant in the previous page figure,

We have

$V_x = V \sin \theta_1$ and $V_y = V \cos \theta_1$

The circular interpolation (say) is in one of the forms

```
G00 X-13 Y112
G02 X-06 Y123 R 109 F 200
```

OR

```
G00 X-13 Y112
G02 X-06 Y123 I40 J23  F 200
```

OR

```
G00 X-13 Y112
G02 X-06 Y123 CX 0  CY 0 F 200
```

That is – we are providing the information about the circle either by the initial point coordinates , final point coordinates and radius
OR
Initial point coordinates, final point coordinates and the coordinates I and J of the initial point with respect to centre of circle taken as origin (without sign)
OR
Initial point coordinates, final point coordinates and centre coordinates.

We understand that at this stage – we need to identify two numbers that we need to put into the X and Y adders that when added to itself repeatedly would produce overflow rates which are proportional to the X and Y velocities.

Now

$V_x = V \sin \theta_1$ and $V_y = V \cos \theta_1$

So we need to get two numbers which are proportional to the sin and cos of the initial angle θ_1

The values I and J are nothing but those very values

$I = R \cos \theta_1$ and $J = R \sin \theta_1$

So simply put these numbers I and J in the Y and X adders respectively and you will get the correct ratio between X and Y velocities at the start of the circular interpolation

Now that the intepolator clock frequency is established as $f = \frac{2^{m+n}}{10}$ and the FRN for linear interpolation is established as 10 V/L, we need to establish the FRN in case of circular interpolation. Is it the same as that of linear interpolation ?

So we can write, for circular interpolation

$$\left\{\frac{f \times FRN \times J}{2^{m+n}}\right\}^2 + \left\{\frac{f \times FRN \times I}{2^{m+n}}\right\}^2 = V^2$$

Noting that $f = \frac{2^{m+n}}{10}$

$$\frac{FRN}{10} R = V$$

Hence

$$FRN = \frac{10V}{R}$$

So far so good

But what about the fact that the Vx and Vy velocities are constantly changing their values ?

Let us see what are their rates of change

Let us look at the scene in a different manner – if we can online update the values of I and J in the adding circuits, we need not bother about anything else and everything will take care of themselves.

Writing I and J as

$I = R \cos \omega t$ $J = R \sin \omega t$

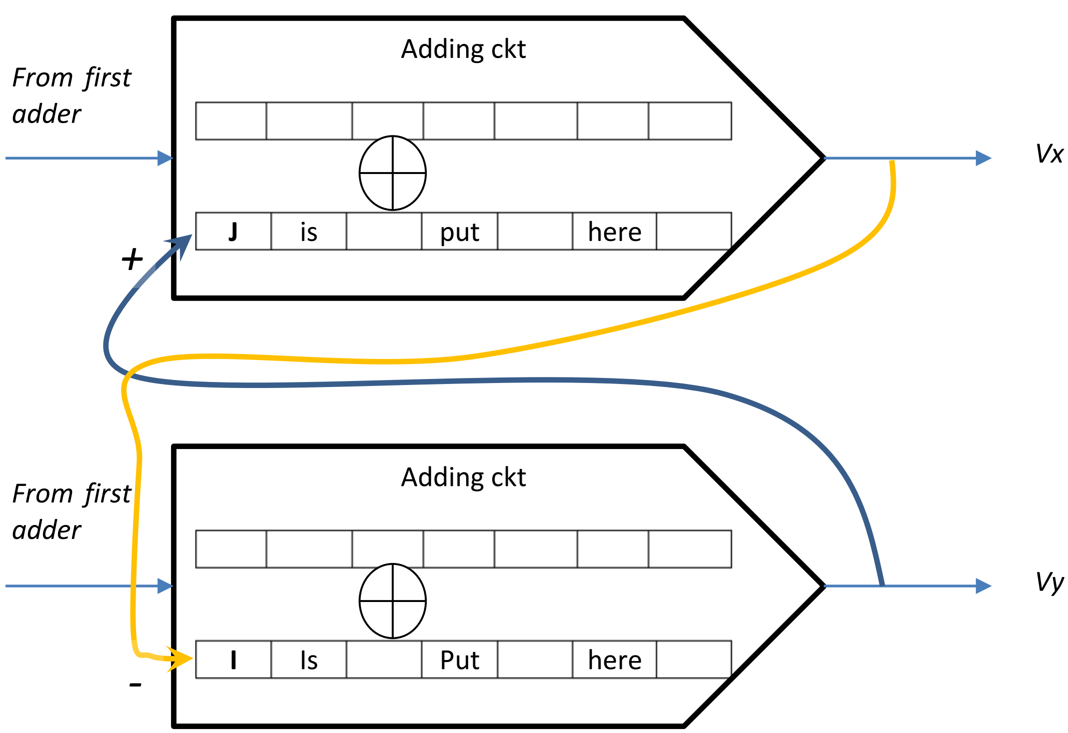
$\frac{dI}{dt} = -R\omega \sin \omega t$ or $\frac{dJ}{dt} = R\omega \cos \omega t$

Incidentally, you will notice that

Vx = V sin ωt = $\frac{dI}{dt}$

Vy= V cos ωt = $\frac{dJ}{dt}$

So the rates at which I and J are supposed to change are nothing but the rates at which the X and Y adders are emitting pulses for Vx and Vy. So we utilize these output pulse rates to update the I and U values. The I and J values are kept inside counters which are having up counting and down counting points (like the ones that we have studied in connection with control loops) and we utilize them as follows :



With this configuration it works fine – at least for this quadrant for circular clockwise cuts. We have not discussed the other configurations and hence they are not in the syllabus.