Autodesk Inventor Tutorials

by Sean Dotson
www.sdotson.com
sean@sdotson.com

Advanced Motion
Part One

Latest Revision: 5/06/02

This tutorial assumes that the user is familiar with basic constraints and how to drive those constrains. It also assumes that they are familiar with creating and editing parameters. If you are not familiar with these topics the help in Inventor is adequate information to understand this tutorial.

To see an .avi file of the final animation lesson click **here**. Now that we know what we want to do let's begin the lesson.

We begin with a basic assembly. The assembly consists of a track, two cylinder housings, two cylinder rods, a cube and a disk. You will find the parts in the zip file that came with this tutorial. Start a new assembly (English) and insert the track. It will be grounded by default. If not, ground it now.

Next use an insert constraint to place the disk in the depression in the track

Constrain the cube to mate against the face of the track and to be flush with the two sides of the track as shown in Figure 1.

At this time the air cylinder body and rod are not necessary. We will use them later in the lesson.

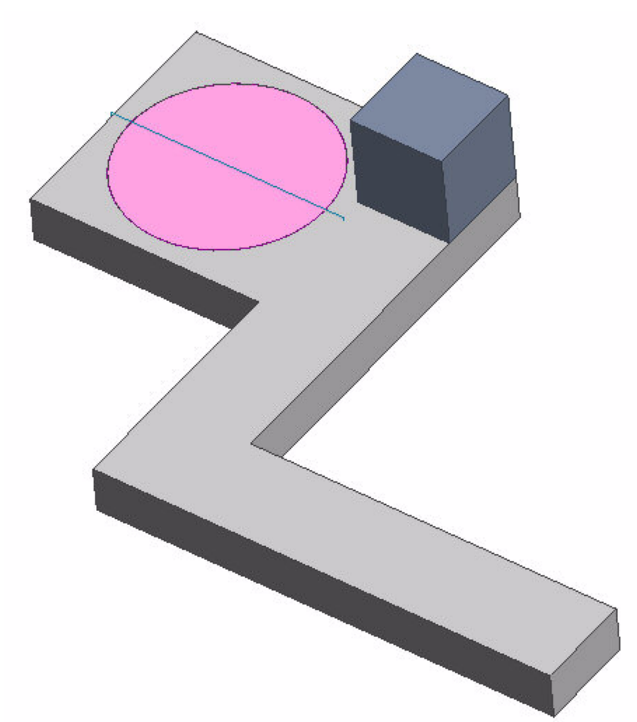


Figure 1 - Basic Constrained Assembly

The basic key to multiple part (or multiple direction motion) is the use of functions. Since Inventor only has the ability to drive one constraint at a time we need to make the motion of the desired parts be driven off the motion of the driven part. In many instances a rotating disk is the best way to have this "base motion".

In out example we will be using the pink disk to drive all motion in the assembly. Constrain the origin plane on the disk with an angle constrain to any parallel surface. (You can also constrain it to one of the assembly origin planes.)

We now want the cube's motion to be derived from the rotation of the disk.

Find the flush constraint that you used to constrain the block to the back side of the track. Place you cursor over the constraint's icon in the assembly browser. The constraint parameter # should appear in a floating help box near the icon. It will be d# (the # will change depending on the order in which things were constrained).

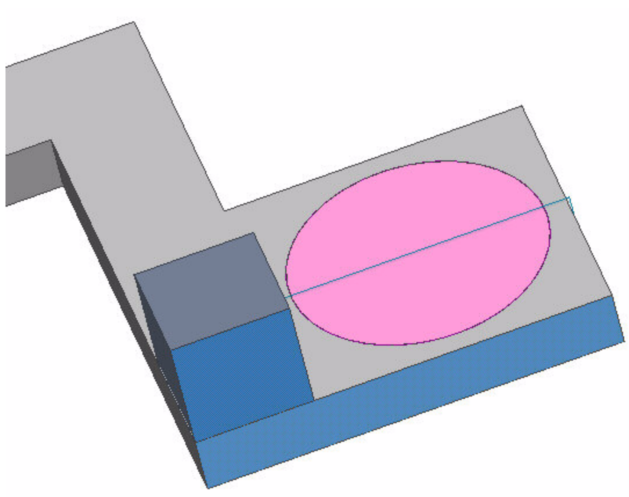


Figure 2 - Flush Constraint # 1

Let's say this constraint was **d5**. Now we also need to know the constraint number for the angular constraint we applied to the disk. Again, hover over the constraint and record the parameter number. Let's say this was **d26**.

Note: Your constraint numbers will likely be much lower than these as in the development of this tutorial a lot of background work was done prior to the writing of this tutorial.

Now open the parameters menu and find parameter **d26** (or whatever you angular constraint was). It should have a value of 0 deg (since it's parallel to the track or origin plane). Now find **d5**. It should have a value of 0 in as it's flush to the track. We want this constraint to vary as a function of the disk's rotation.

Let's first rename these parameters to something a bit more useful. Rename **d26** (angular) to **angle**. Rename **d5** (flush) to **move1**.

Refering back to the lesson on Inventor Function we can see that we want to use the min/max functions to make the block move.

We want the block to move to the end of the track (a total distance of 4 in) and then stop movement in that direction.

We can now tie the value of the **move1** constraint to the value of the **angle** constraint.

Edit the equation of **move1** to read:

```
min(0.100 ul * angle * 1.000 in / 1.00 deg;4.000 in)
```

Let's break down this equation to understand what it's doing. The equation will take the minimum value of either 0.1***angle** or 4 in. Since the value of **angle** will be varied it is constantly changing. Therefore the value of **move1** will change as long as 0.1***angle** is less than 4 in.

Note: You must keep units consistent. Since move1 is a linear constraint and has units of inches you must multiply angle by 1.0 in/ 1.0 deg to change the value of the angle into a linear unit. You could also use the isolate function but for simplicity we have chosen to do it the mathematical way. If your equation is red then these is an error. Most times it is do to inconsistent units.

We chose to multiply angle by 0.1 as a scaling factor. This means angle will have to grow to 40deg (0.1 * 40 = 4) before the motion stops. You can vary this scaling factor to suit your needs.

Now exit the parameters menu. Find the **angle** constraint in you browser and drive it from 0 to 50 deg. The cube should move from its starting position until it reaches the end of the track (4 in) and then stop. You can continue to drive **angle** to infinity and the cube will not move past the 4 inches. (see Figure 3)

Reset the cube it it's initial position by driving it back to 0 deg and apply the angle.

We now want the cube to move to the end of the track in the other direction **after** making the move we have just programmed. (Follow the track)

We can do this by varying the second flush constraint that we set up when building the assembly. Find the number of this parameter. Enter the parameter menu and change its name to **move2**. (see Figure 4)

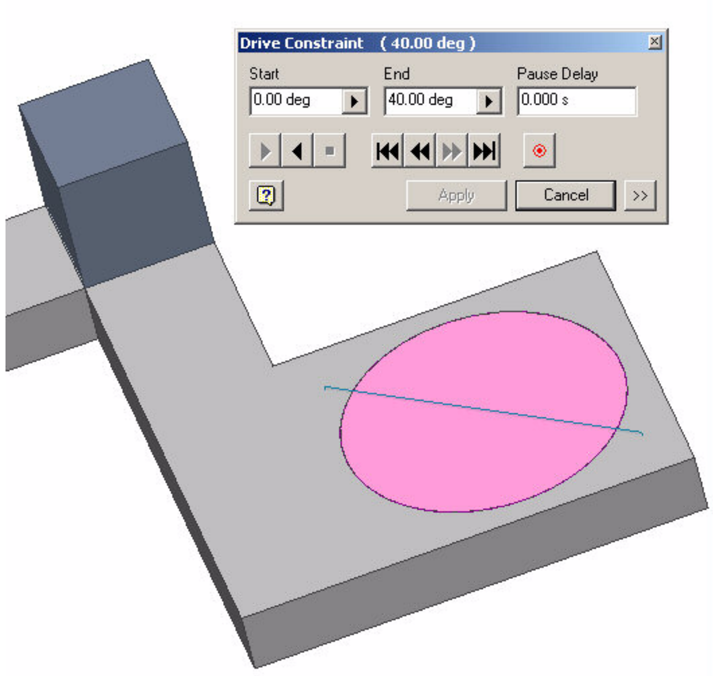


Figure 3 - Motion Stops at Angle = 40 deg

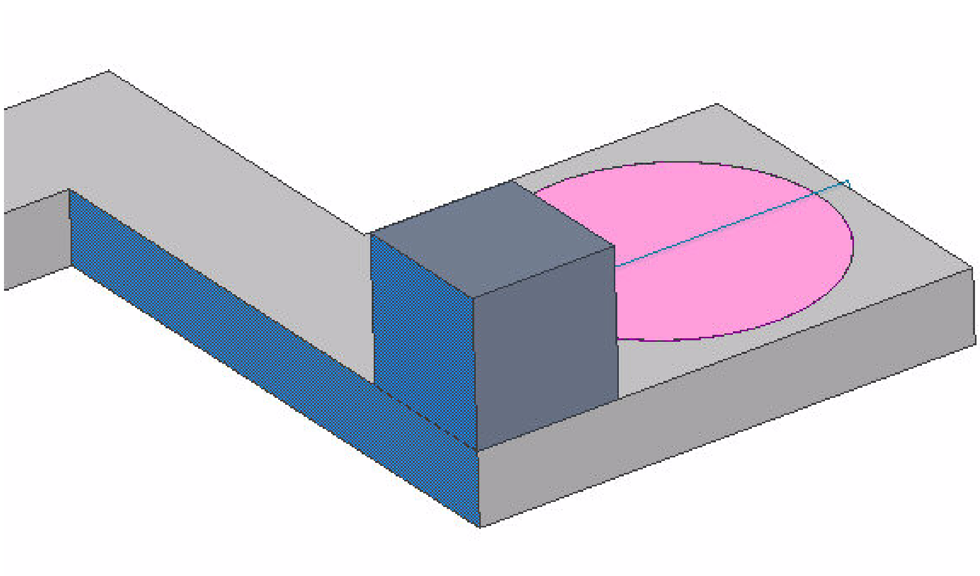


Figure 4 - Second Flush Constraint

We now want to make **move2** vary with **angle**. Enter the parameters menu and edit **move2**'s equation to:

```
max(0 in;( ( ( angle * 1 in / 1 deg ) - 60 in ) * 0.1 ul ))
```

Again we shall break down the equation. Move2 will be equal to the larger of the values of 0 in or (angle –60)*0.1 This is set up a bit differently than the first equation. We know we want **move2** to be 0 from angle values of 0 to 60 deg hence the 0 in in the max equation. *(We chose 60 to have a delay between direction changes. This value could be anything greater than or equal to 40).* Then we want it to move a value equal to (angle-60)*.1 for the remainder for the values of **angle**. Again 0.1 ul is used as a scaling factor. We could have changed this to 0.5 ul to make it appear that the second move is faster than the first.

Exit the parameters menu and then drive **angle** from 0 to 120 degrees. The block should move along the first path to 40 degrees, remain stationary from 40 to 60 degrees and then continue along the second path from 60 to 120 degrees. Notice that the block goes off the end of the path after 100 degrees. There are two way to prevent this. You can either only drive **angle** to 100 degrees (simple method). Or you can edit the equation from move2 to make the block stop after 100 degrees of motion.

This technique will be covered in the second part of the motion tutorial.

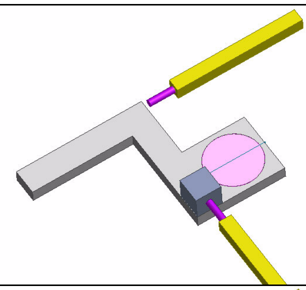
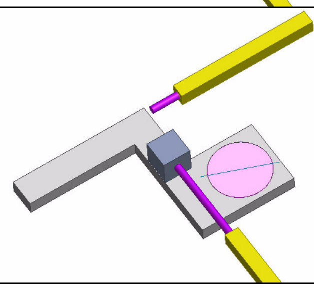
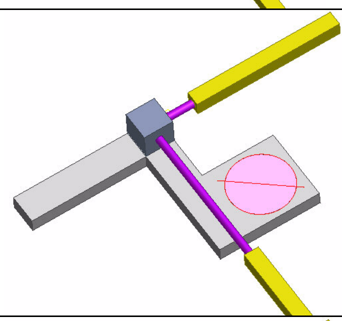
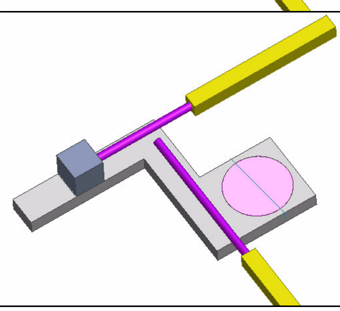
Perhaps the trickiest part of writing good motion equations is matching up the parentheses. It can be quite difficult to make sure you have the correct number and in the correct position to obtain the desired effect. On tip is to use Excel to write you equation. Excel will highlight each set of parentheses as you type to make sure you have them matched properly. As an added bonus Excel has min and max equations as well. You can plan you motion and experiment in Excel before writing the equations in Inventor.

If you want to take it a step further you can actually write you equations in Excel and then link or embed then into the parameters table. Then use your user parameter names in the model parameters table.

Now to dress up our assembly we can insert the cylinder bodies and rods and constrain them in such a manner than the body has 0 degrees of freedom (effectively grounded) and the rods are mated axially to the bodies and with a mate constraint to the faces of the block. This gives the illusion that the cylinders are pushing the block along the path. See the .avi file located at **http://www.sdotson.com/**.

The final assembly should look like the figures below.

In the next tutorial we shall look at driving multiple parts off one parameter and more advanced uses of Inventor functions.

Value of angle	Figure
angle=0	
40>angle>0	
40>angle>60	
angle>60	
angle=100