Laser Raster-Scanning Patterns on Cylindrical Workpieces Using CADFusion and PSO Bitmap Tools

By: Scott Schmidt
Applications Group Manager

A large number of cylindrical laser processing applications program the laser writing campaign as if the two axes (one linear and one rotary) are both, in fact, linear in nature. In other words, the cylindrical pattern is "unrolled" to make the shape a more typical flat workpiece. While this technique might ease the motion control programming burden, often the resulting pattern cycle time suffers due to the many directional changes the rotary axis (in particular) must undergo during laser processing. Alternatively, "free-running" the rotary axis in a constant direction and raster-scanning the required pattern using high-dynamic galvanometers is a better and far more time-efficient means of writing the same pattern to the cylindrical workpiece.

Figure 1 shows a typical unrolled stent pattern. Because of the many tight circular features in the pattern, both axes must accomplish many directional changes while executing the motion profile. As workpieces become smaller in diameter, the acceleration values implied by these directional changes become very high. At certain accelerations the stages can no longer effectively track the desired profile and cycle times must be increased (as acceleration values are lowered) to compensate.

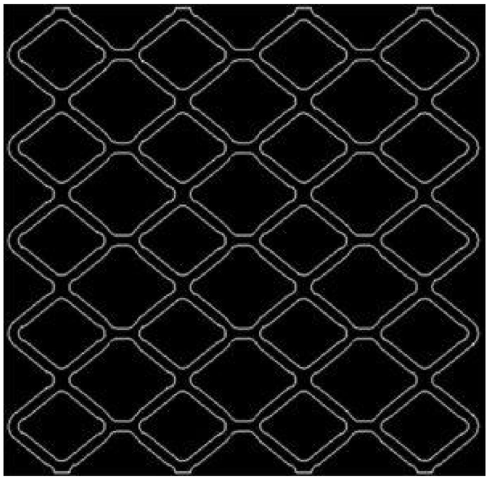


Figure 1. *Stent pattern "unrolled."*

However, if the image shown in Figure 1 were sampled to create pixels whose sizes are an integer multiple of the laser spot size, then the entire pattern could easily be written using a raster style from the resulting bitmap array. Furthermore, CADFusion and the PSOOUTPUT BITMAP and Mark-on-the-Fly features native to the Aerotech Automation 3200 (A3200) controller can facilitate writing the pattern onto a continuously rotating workpiece using a high-speed galvanometer (Aerotech's Nmark AGV product line). One axis of the AGV performs the linear scanning while the second axis compensates for the rotation of the workpiece during a single raster line – i.e., making sure that the line is not diagonal on the workpiece. Of course, care must be taken to choose a rotational speed and scanning velocity that will not cause the laser spot to "walk off" the workpiece and become defocussed.

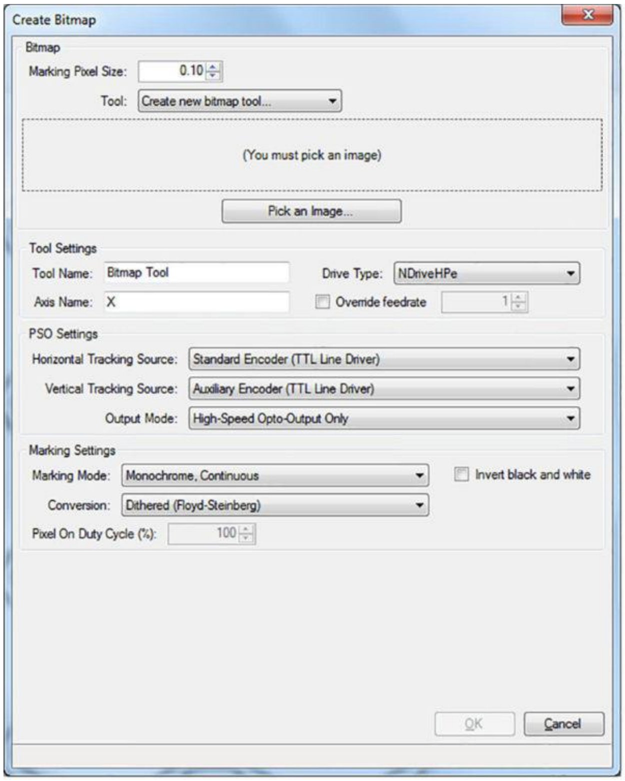
The steps required to create such a project follow.

Create BitMap from Drawing

First, save the source DXF file to a bitmap format of highest possible resolution. Ensure that the pixel size is scaled or chosen to be an integer multiple of the laser spot being used to process the part. Other sampling strategies might include simple snipping/screen captures (for lower resolution trials) or third-party programs to convert vector files to bitmap images.

Use CADFusion to Create Bitmap Project

Next, open CADFusion and begin a New Project. The first key step is to import a Bitmap object onto the canvas. This is done by selecting the Bitmap item from the Create Group within CADFusion. The resulting Dialog Box is shown in Figure 2.



Bitmap

Marking Pixel Size: 0.10

Tool: Create new bitmap tool...

(You must pick an image)

Tool Settings

Tool Name: Bitmap Tool

Drive Type: NDriveHPe

Axis Name: X

□ Override feedrate 1

PSO Settings

Horizontal Tracking Source: Standard Encoder (TTL Line Driver)

Vertical Tracking Source: Auxiliary Encoder (TTL Line Driver)

Output Mode: High-Speed Opto-Output Only

Marking Settings

Marking Mode: Monochrome, Continuous

□ Invert black and white

Conversion: Dithered (Floyd-Steinberg)

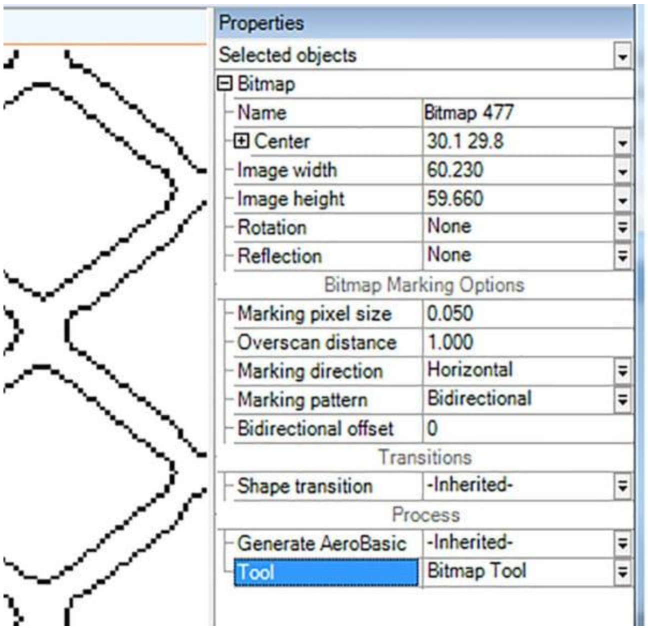
Pixel On Duty Cycle (%): 100

Figure 2. *Bitmap dialog within CADFusion.*

The desired bitmap image should be selected (using the "Pick an Image…" button), and the Bitmap Tool Settings should be chosen. CADFusion allows easy selection of the drive hardware which simplifies Tracking Source (from where the servo encoder signals come for each of the two axes) as well as Marking Settings. For our example, we will check the "Invert black and white" box (as our bitmap was captured on a black background canvas), and we will select the "Threshold, Bitmap's Average Pixel Value" choice from the Conversion dropdown list. Furthermore, we will maintain the default Drive Type and PSO Settings, and we will also acknowledge that the first axis – which also will issue the PSO pulses to the laser – is named X within the A3200 controller parameter file.

After the Bitmap settings have been made, CADFusion allows the user to choose where to place the actual Bitmap Object on its canvas. We will manually type "0, 0" to place the bottom left-hand corner of the bitmap at the origin of CADFusion's canvas. Notice that after the bitmap has been placed, the Bitmap Tool has been created and is visible under the project tree on the left-hand side of the CADFusion window.

Next, we must define the marking pixel size within the Bitmap Object and Tool. Clicking on the bitmap image on the canvas or on the project tree exposes its properties on the right-hand side of the CADFusion window (Figure 3).



Properties

Selected objects

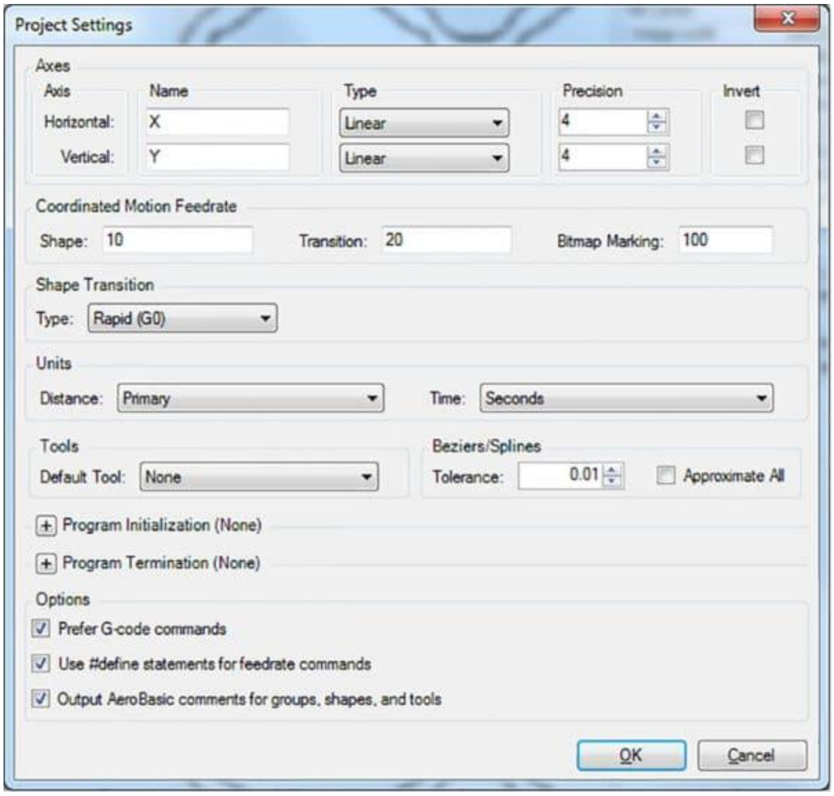
⊟ Bitmap

Name	Bitmap 477
⊞ Center	30.1 29.8
Image width	60.230
Image height	59.660
Rotation	None
Reflection	None
Bitmap Marking Options	
Marking pixel size	0.050
Overscan distance	1.000
Marking direction	Horizontal
Marking pattern	Bidirectional
Bidirectional offset	0
Transitions	
Shape transition	-Inherited-
Process	
Generate AeroBasic	-Inherited-
Tool	Bitmap Tool

Figure 3. *Bitmap properties.*

Within the Properties panel we select a Marking pixel size of 0.050, which corresponds to 50 microns for the Configuration file we intend to use within A3200 (and is the spot size for the laser and optical setup which will be used). In other words, a User Unit within the A3200 parameter file is one millimeter.

The final step within CADFusion is Exporting the actual A3200 code. Select Export from the File Menu Group. This launches the Project Settings dialog if these have not already been previously set (Figure 4).



Project Settings

Axes

Axis	Name	Type	Precision	Invert
Horizontal:	X	Linear	4	□
Vertical:	Y	Linear	4	□

Coordinated Motion Feedrate

Shape: 10 Transition: 20 Bitmap Marking: 100

Shape Transition

Type: Rapid (G0)

Units

Distance: Primary Time: Seconds

Tools

Default Tool: None

Beziers/Splines

Tolerance: 0.01 □ Approximate All

⊞ Program Initialization (None)

⊞ Program Termination (None)

Options

☑ Prefer G-code commands

☑ Use #define statements for feedrate commands

☑ Output AeroBasic comments for groups, shapes, and tools

Figure 4. *Project settings.*

These settings should be chosen to support the feedrates allowed by the system mechanics. For galvo-enabled marking, the Coordinate Motion Feedrates can be set quite high – easily 100s of mm/s and perhaps higher. The Axis Names should be verified against the controller parameter file, and Linear Type should be selected for both axes (recall that the rotary stage is being programmed as a linear axis in circumferential units for this project). If any Initialization or Termination code is required (for instance, to enable and home the axes, to set up machine safety or I/O functionality, or to make function declarations and definitions), this may be typed or imported within this dialog box.

Once the Project Settings have been selected and saved, the resulting code is displayed and should be saved.

Configure Mark on the Fly

Next, open the saved program within A3200 Motion Composer. To enable the Mark on the Fly feature required for continuous-rotation laser machining, the following command is required to allow the radial-direction galvo axis (Y direction in our example) to have access to the rotary stage's encoder feedback (and to then compensate for it during X-direction scans):

GALVO ENCODERSCALEFACTOR Y (CountsPerUnit.Y / CountsPerUnit.THETA)*EmulatedQuadratureDivider.THETA

In this example the rotary axis is named THETA. This statement not only binds THETA to the Y galvo axis direction, but also scales the relative axes such that one user unit in the galvo direction is equivalent to the same user unit in the servo (THETA) direction.

The EmulatedQuadratureDivider qualifier within this statement is optional and may be omitted if this THETA axis parameter is set to 1 within the configuration file. This statement should be inserted into the program before the "' Bitmap 1" line comment to be certain that it is implemented before any motion is commanded of the system.

The specific wiring for the system should also be confirmed at this time to be certain that the encoder feedback signals from the rotary stage's drive are properly input into the galvo drive's AUXIO port.

From the A3200 Help File

Connecting the Encoder Feedback Cables

Use the correct cable to connect the output of the servo drive to the auxiliary input of the galvo drive (Nmark).

Servo Drive Type	Cable Part Number
Npaq	C23320-10
Ndrive	C23161-xx

Table 1. *Servo drive cable part numbers.*

After you configure the parameters and commands and you connect the servo drive to the Nmark, verify the configuration before you try to run a program in IFOV mode.

1. Open Motion Composer.
2. Open Status Utility and click the Diagnostics tab.
3. Enable the servo axes.

4. In Motion Composer, jog the servo axes and look at the Aux Pos Fdbk signal on the Nmark axes in Status Utility.
5. Verify that the Aux Pos Fdbk is counting in the same direction as the servo axis.
    a. If the counter is not changing, then there is a problem with configuration or wiring.
    b. If the counter is counting in the opposite direction as Pos Fdbk of the servo axes, then the wiring polarity is inverted.

The final step in configuring this feature is to command the rotary stage to freerun. In general this would be done immediately before galvo motion is commanded to write the actual part. However, CADFusion outputs a very compact code structure for bitmap generation that nests the motion commands within those commands that download the bitmap arrays to drive memory. Therefore the required insertion point for the FREERUN command may not be obvious. This command:

```
FREERUN THETA <feedrate>
```

Should be inserted immediately following the line within the CADFusion-generated code:

```
$nextReadScan = $nextReadScan + $maxScansPerRead
```

Similarly, command:

```
FREERUN THETA STOP
```

Should be inserted toward the end of the program after command line:

```
FILECLOSE $BitmapDatFile
```

This stops the rotary axis after the laser processing is complete.

One key factor to consider is the feed rate chosen for the rotary stage. If a feed rate too slow or too fast is chosen, the galvo axis in the radial direction (again, Y in our example) will still track the rotary stage's motion, but it may cause the laser spot to "fall off" to one side or the other of the cylindrical workpiece as it tracks.

To a first order approximation the total processing time may be estimated as the number of raster scans – i.e., image length divided by pixel size – times the product of scan speed (Coordinated Motion Feedrate) and image width:

Process Time = (Image Length/pixel size) * (Coordinated Motion Feedrate) * (Image Width)

Of course this assumes that the galvo axis in the axial direction (X) can accelerate in near-zero time. All dimensions are in user units, and the Coordinated Motion Feedrate should be replaced with any explicitly commanded RATE value used for that axis in the program. The feedrate for the FREERUN command should be selected to support one revolution of the rotary axis in this amount of time.

Summary

The preceding example and supporting code demonstrate a feature unique to the A3200, and one that is enhanced by the powerful image post-processing capabilities of CADFusion.

About the Author

Scott Schmidt is an applications engineering manager at Aerotech Inc., where he has gained 17 years of experience with advanced laser processing and micromachining. He holds a bachelor's degree in electrical engineering from Penn State University and a master's degree in electrical and computer engineering from the University of Massachusetts.