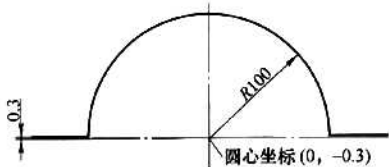


半圆数控程序的误差分析

南阳中光学集团精密机械制造有限公司 (河南 473006) 胡飞嘉

编写半圆的数控程序时很容易产生误差的问题。半圆数控程序的误差主要来自于运算精度而不是机床本身的运动精度。程序中数值的运算精度至少能达到 10^{-8} ，数控程序能接受 3 位小数的输入。而数控机床的运动精度通常在 $0.004 \sim 0.01\text{mm}$ 左右。如此看来，数值的运算精度远高于机床的运动精度，似乎是只要程序中的数值不输错，就不会产生对零件形状有影响的误差。实则不然！下面用一个实例来说明这个问题。

数控编程编直线程序段时，插补产生的中间点的误差肯定不会超过端点的误差。然而在编半圆程序时，则会产生非常大的误差。在制图软件中画一个半径 100mm 的圆，圆心位于 $(0, -0.3)$ ，再从 $(-150, 0)$ 到 $(150, 0)$ 画一条直线。然后把圆的下半部和直线的中部修剪掉，结果如图所示。



显然，剩下的圆弧是小半圆（因为直线在 X 轴上，而圆心则在 X 轴下方 0.3mm 处）。如此简单的形状，编写数控程序就会产生很大的误差。用 MasterCAM 生成的数控程序如下：

```
G0 X -150 Y0
...
G1 X -100
G2 X100 R100
G1 X150
...
```

程序很简单，也很直观。可是仔细分析却发现，程序中的圆弧是从 $(-100, 0)$ 到 $(100, 0)$ 的半径为 100mm 的圆弧，这不恰好是个半圆吗？半圆的

圆心在 $(0, 0)$ 。而原始图形的圆心却在 $(0, -0.3)$ 。也就是说，圆弧的最高点产生了 0.3mm 的巨大误差！如此大的误差显然是不能满足生产需要的。是程序出错了吗？逐点检查程序，确实没有出错。那么这么大的误差是怎么产生的呢？

原来，问题出在圆弧两端点的非常微小的舍入误差上。既然已经知道原始图形的圆弧是小半圆，则左端点的 X 坐标不可能是 -100 。把制图软件的小数位数设在 5 位以上，查询圆弧端点，可以发现，左端点的坐标是 $(-99.99955, 0)$ ，右端点的坐标是 $(99.99955, 0)$ 。然而数控机床只接受 3 位小数，生成的数控程序也只保留 3 位小数，于是 99.99955 被四舍五入成 100.000 。这就是问题产生的原因。

从表面上看，数控机床接受 3 位小数已经足够了，毕竟产生的舍入误差还不到 $0.5\mu\text{m}$ 。可是从这个例子就可以看出，圆弧接近 180° 时，产生的误差是非常大的。圆弧半径越大，产生的误差也越大。无论是手工编程还是 CAM 软件自动生成，均会产生这种误差。而且 MasterCAM 软件在生成大半圆的程序时，通常是把半圆生成一程序，把剩下的一小段圆弧再生成一程序，从而增加了出现此问题的几率。在生成整圆时如分成两个半圆则更容易出现。

怎样才能避免出现此问题呢？在编写接近 180° 的圆弧的数控程序时，可把圆弧分成两端来编程，也可以用 I、J 而不用 R 来编程（上述例子中的 G2 那一程序改成 G2 X100 I100 J-0.3），就可以避免产生这样的问题。在编写大半圆的数控程序时，不要分成两段，直接用一程序表示（R 为负值时表示大半圆）；在编写整圆的程序时，用 I、J 表示而不要分成两个半圆，也不会产生这样的问题。

正所谓“失之毫厘，谬以千里。”半圆数控程序产生误差的问题值得注意！MW

(收稿日期：20101214)