

京微雅格企业杯：FPGA 设计

4. 基于 FPGA 圆弧插补算法的设计

B194

大连理工大学 alphago 队

李述博 安振霖 关桑海

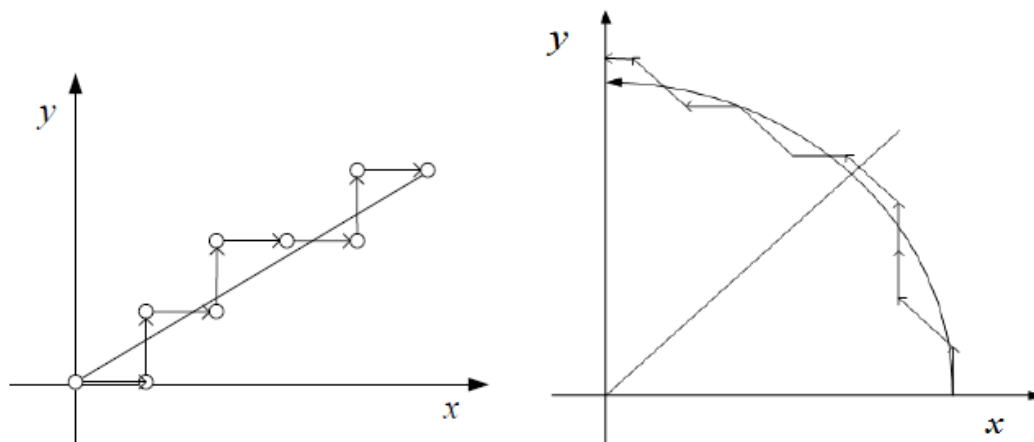
指导教师：王开宇

目录

1. 设计原理	2
1.1 逐点比较法	2
1.2 最小偏差法	4
1.3 比较积分法	6
2. 设计流程	8
2.1 算法模型向电路结构抽象转换	8
2.2 电路结构设计和功能模块划分	8
2.3 实际电路设计	9
2.4 设计验证以及改进	9
3. 可能遇到的问题以及预期解决办法	10
4. 最终成果预测	11
4.1 算法的软件仿真	11
4.2 实际硬件实现	11
4.3 用户接口	12

1. 设计原理

在数控机床中，刀具或工件的最小位移量是机床坐标轴运动的一个分辨单位，由检测装置辨识，称为数控系统的分辨率或脉冲当量，因此刀具的实际运动轨迹微观上是由一系列脉冲当量段构成的折线。从下图中可知，插补过程就是使用机器所识别的最小单位，组成曲线和曲面的过程。在组成过程中采用的算法即为插补算法。

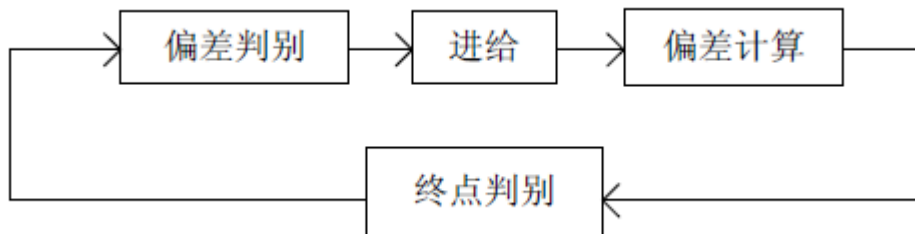


直线插补与圆弧插补示意图

插补主要分为粗插补和精插补两部分。粗插补计算获得每个插补周期内刀具或执行末端应移动的长度，然后将这段距离向各个坐标轴投影，可获得该插补周期内各轴的增量。将这些增量依次在相同的时间内(一个插补周期)转化成相应的脉冲和方向发送到全数字伺服系统，便可完成轮廓或曲线的加工，即精插补过程。因此，精插补过程是将每个插补周期获得的刀具或末端移动长度通过算法转为控制的脉冲和方向的过程，常见的几种圆弧插补算法主要有以下几种。

1.1 逐点比较法

逐点比较法是通过偏差判别、进给、偏差计算、终点判别四个步骤循环执行进行的，其主要流程如下：



逐点比较法示意图

从上面的流程知，关键是偏差计算，即确定偏差函数。针对不同线性插补，有不同的偏差函数，圆弧的偏差函数计算过程如下：针对第一象限逆时针圆弧，其圆弧段上，加工点坐标为 (x_n, y_n) ，终点坐标 (x_z, y_z) 。则加工点的偏差函数为：

$$F_n = x_n^2 + y_n^2 - R^2$$

若 $F_n \geq 0$ ，加工点位于圆弧外侧；若 $F_n < 0$ ，加工点位于圆弧内侧。这样计算偏差极不方便，且不易用计算机实现；因此，仍采用递推法得到相邻两点之间偏差函数的递推式。

还以圆心位于坐标原点的第一象限逆时针圆弧为例，假设某时刻，加工点位于圆弧之外，即 $F_n \geq 0$ ，为了减小偏差，应向 $-\Delta x$ 方向走一步，则新加工点坐标为 $x_{n+1} = x_n - 1$ ， $y_{n+1} = y_n$ ，则新的偏差函数为：

$$\begin{aligned} F_{n+1} &= x_{n+1}^2 + y_{n+1}^2 - R^2 \\ &= (x_n - 1)^2 + y_n^2 - R^2 \\ &= F_n - 2x_n + 1 \end{aligned}$$

同理，若加工点位于圆弧内侧，应向 $+\Delta y$ 方向走一步，此时，新的加工点坐标为 $x_{n+1} = x_n$ ， $y_{n+1} = y_n + 1$ ，则新的偏差函数为：

$$\begin{aligned} F_{n+1} &= x_{n+1}^2 + y_{n+1}^2 - R^2 \\ &= x_n^2 + (y_n + 1)^2 - R^2 \\ &= F_n + 2y_n + 1 \end{aligned}$$

可知：圆弧新加工点的偏差函数式 F_{n+1} 是由前一点的函数式 F_n 而来的，它不仅要用前一点的函数式值，还要利用前一点的坐标值 x_n 或 y_n ；所以计算偏差函数时，不但需要保存 F_{n+1} ，还要保存动点的坐标值 x_{n+1} 或 y_{n+1} ，以便下一点偏差计算使用。

针对圆弧不同象限和不同进给方向，偏差计算也不相同。记 R 代表圆弧，S 代表顺圆，N 代表逆圆，L 代表直线，下标阿拉伯数字代表象限，则圆弧可分为八种情况： $SR_1, SR_2, SR_3, SR_4, NR_1, NR_2, NR_3, NR_4$ 来表示；直线分为四种情况： L_1, L_2, L_3, L_4 ，具体偏差计算下表。

线型	$F \geq 0$ 进给方向	$F < 0$ 进给方向	偏差计算
SR_1	$-\Delta y$	$+\Delta x$	$F \geq 0$ 时,
SR_3	$+\Delta y$	$-\Delta x$	$C_3: F - 2y + 1 \rightarrow F$ $y - 1 \rightarrow y$
NR_2	$-\Delta y$	$-\Delta x$	$F < 0$ 时,
NR_4	$+\Delta y$	$+\Delta x$	$C_4: F + 2x + 1 \rightarrow F$ $x + 1 \rightarrow x$
SR_2	$+\Delta x$	$+\Delta y$	$F \geq 0$ 时,
SR_4	$-\Delta x$	$-\Delta y$	$C_1: F - 2x + 1 \rightarrow F$ $x - 1 \rightarrow x$
NR_1	$-\Delta x$	$+\Delta y$	$F < 0$ 时,
NR_3	$+\Delta x$	$-\Delta y$	$C_2: F + 2y + 1 \rightarrow F$ $y + 1 \rightarrow y$
L_1	$+\Delta x$	$+\Delta y$	$F \geq 0$ 时,
L_2	$-\Delta x$	$+\Delta y$	$C_5: F - y_z \rightarrow F$
L_3	$-\Delta x$	$-\Delta y$	$F < 0$ 时,
L_4	$+\Delta x$	$-\Delta y$	$C_6: F + x_z \rightarrow F$

不同象限偏差计算

1.2 最小偏差法

对于圆弧，一般圆弧中心作为坐标原点。设若加工第一象限圆弧 QZ，半径为 R，方向由 Q ~ Z。则进给方式为：当 $x > y$ 时，进给方向为 y 方向或 x, y 方向同时进给；当 $x < y$ 时，进给方向为 x 或 x, y 方向同时进给。进而研究圆弧的偏差判别，因圆弧方程为：

$$x^2 + y^2 - R^2 = 0$$

设加工点坐标为 M (x_M, y_M)，由 $x_M^2 + y_M^2$ 与半径 R 相比较，便可知道加工点的位置。则偏差函数为：

$$F(x, y) = x_M^2 + y_M^2 - R^2$$

而加工点 M(x_M, y_M) 到圆弧的径向距离 ΔR 为：

$$\Delta R = \left| \sqrt{x_M^2 + y_M^2} - R \right|$$

$$\begin{aligned}
&= \frac{\left| \sqrt{x_M^2 + y_M^2} - R \right| \cdot \left| \sqrt{x_M^2 + y_M^2} + R \right|}{\left| \sqrt{x_M^2 + y_M^2} + R \right|} \\
&= \frac{x_M^2 + y_M^2 - R^2}{\left| \sqrt{x_M^2 + y_M^2} + R \right|} = \frac{|F(x, y)|}{\left| \sqrt{x_M^2 + y_M^2} + R \right|}
\end{aligned}$$

而上式也正给出了偏差值和偏差函数之间的关系。

进给偏差计算，假设加工点 M (x_M , y_M) 的坐标 $x_M > y_M$ ，此时进给方向为 Δy 或对角线方向 ($-\Delta x$, $+\Delta y$ 同时进给方向)。当向 y 轴进给一步，则新的加工点偏差函数 $F(\Delta y)$ 为：

$$\begin{aligned}
F(\Delta y) &= x_M^2 + (y_M + 1)^2 - R^2 \\
&= x_M^2 + y_M^2 + 2y_M + 1 - R^2 \\
&= F(x_M, y_M) + 2y_M + 1
\end{aligned}$$

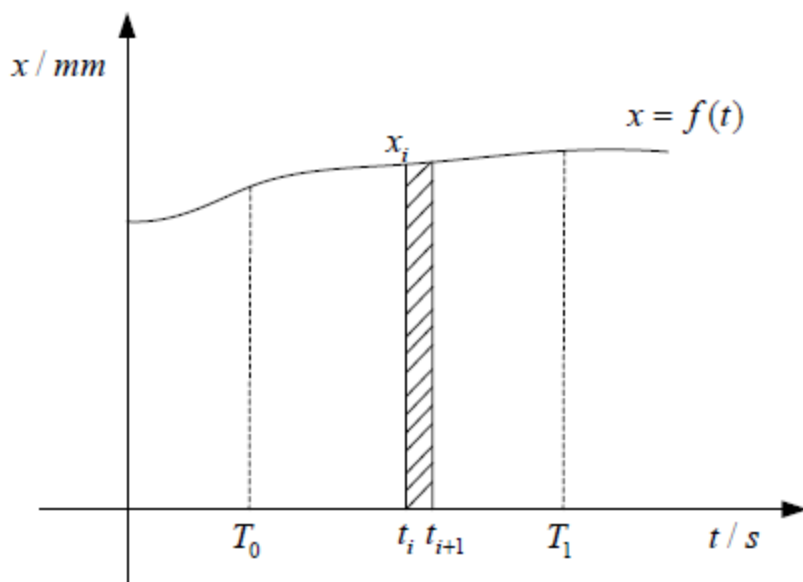
当向对角线方向进给一步，则得到另一新的偏差函数 $F(\Delta x, \Delta y)$ 如式：

$$\begin{aligned}
F(\Delta x, \Delta y) &= (x_M - 1)^2 + (y_M + 1)^2 - R^2 \\
&= F(x_M, y_M) - 2x_M + 2y_M + 2
\end{aligned}$$

比较 $|F(\Delta y)|$ 和 $|F(\Delta x, \Delta y)|$ 的大小，选择其中较小的代表的方向进给，即：若 $|F(\Delta y)| \leq |F(\Delta x, \Delta y)|$ ，则进给 $+\Delta y$ ；若 $|F(\Delta y)| > |F(\Delta x, \Delta y)|$ ，则进给 $(-\Delta x, +\Delta y)$ 。其余象限和顺逆圆弧进给方式计算类似。

1.3 比较积分法

数字积分法 (Digital Differential Analyzer) 不需要复杂的判别函数及计算，运算速度快，适用于实时性的运动控制。



数字积分法原理图

在平面内坐标系内，任意函数 $x = f(t)$ ，如上图所示，而从时间 T_0 到 T_1 函数 $f(t)$ 下包围的面积为：

$$S = \int_{t_0}^{t_1} f(t) dt$$

将时间 T_0 到 T_1 分割成 m 个比较小的时间段， m 趋近无穷大时，上式可近似为：

$$S = \sum_{i=0}^{m-1} x_i \Delta t = \Delta t \cdot \sum_{i=0}^{m-1} x_i$$

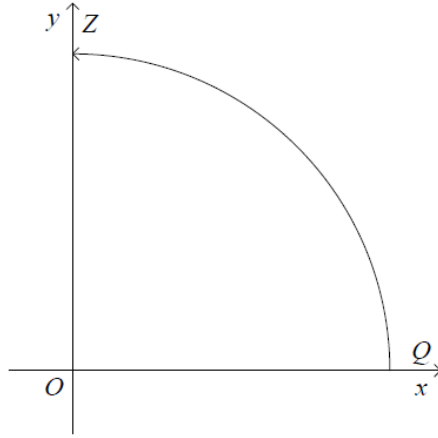
式中 $m \cdot \Delta t$ —— 为 $t_1 - t_0$ ；

x_0 —— 为 $f(t_0)$ ；

x_m —— 为 $f(t_1)$ 。

可见数字积分法就是将复杂的积分运算转化成简单的累加运算。

加工圆弧时， x 轴的速度正比于动点到 y 轴的距离； y 轴的速度正比于动点到 x 轴的距离。具体推导如下：设第一象限的逆时针圆弧 QZ ，其上的动点 $Q_i(x_i, y_i)$ ，圆弧的加工速度（切向速度）为 V ，如下图所示。



圆弧插补示意图

从上图中，可以得到点 $Q_i(x_i, y_i)$ 处的速度满足：

$$\frac{V}{R} = \frac{V_y}{x_i} = \frac{V_x}{y_i} = k$$

则在 Δt 时间内，动点 $Q_i(x_i, y_i)$ 沿 x 和 y 轴的位移为。

$$\begin{cases} \Delta y = V_y \Delta t = k x_i \Delta t \\ \Delta x = -V_x \Delta t = -k y_i \Delta t \end{cases}$$

而位移增量的方向可以通过象限确定，所以位移增量的绝对值由式求得。

$$\begin{cases} |\Delta y| = k |x_i| \Delta t \\ |\Delta x| = k |y_i| \Delta t \end{cases}$$

所以圆弧插补过程中，动点沿轴的增量比例于其对应的轴距离；且插补过程中需不断更新动点坐标 $Q_i(x_i, y_i)$ 。

2. 设计流程

2.1 算法模型向电路结构抽象转换

这里以积分法为例子，圆弧插补模块由数据缓存器、插补积分器和位置计数器构成。

1. 数据缓存器

数据缓存器共由 6 个 n 位寄存器构成，它们分别存储 x , y , z 三个轴的初始点坐标值及终点坐标值。数据缓存模块将直线插补的数据进行自动加载，以便于插补积分器调用模块内的数据。进行缓存的目的是为了保持数据的流畅性，确保插补不因为数据的转变而出现中断的情况。这样大大地提高插补的效率，同时减少因为时序问题所带来的误差。当积分器加载缓存器中的数据完毕以后，积分器接收到 start 信号，积分器开始进行数字积分直线插补。当数据缓存器接收到 rest 信号时，缓存器中所有数据全部做复位操作。

2. 插补积分器

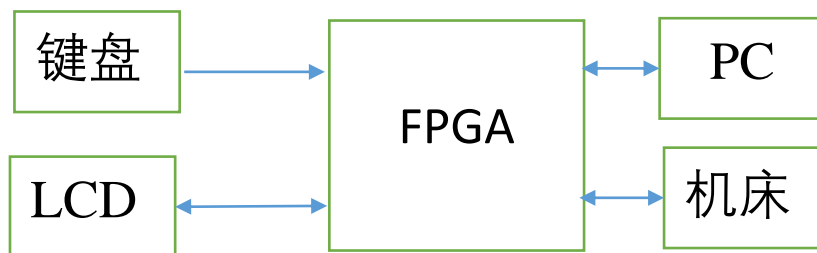
插补积分器是直线插补的核心模块，由余数寄存器和被积函数寄存器构成。被积函数寄存器中存放由数据缓存器提供的终点坐标值，每当脉冲源发出 1 个插补迭代脉冲时，被积函数寄存器与余数寄存器的值累加 1 次，并将累加结果存放余数寄存器中，当累加结果超出余数寄存器容量 (2^n) 时，溢出 1 个脉冲。经过 $N=2^n$ 次累加后，每个坐标轴的溢出脉冲总数就等于该坐标的被积函数值。在余数寄存器的容量一定的情况下，其输出脉冲频率与终点坐标值的大小成正比。控制器采用三轴联动，则同时有 3 个积分器同时进行插补运算。

3. 位置计数器

位置计数器有 3 个寄存器，分别对应于三个轴。其初始值为需要插补的脉冲数目。当积分器每输出一个脉冲，计数器便做减 1。当位置计数器为零时，则表示到达终点坐标位置，插补结束。

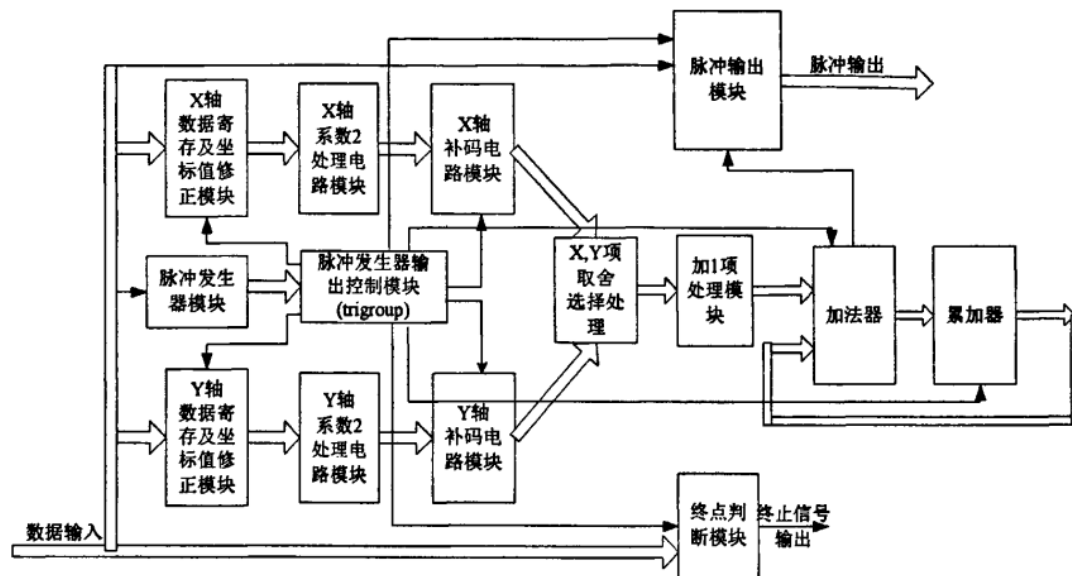
2.2 电路结构设计和功能模块划分

整体结构：



预期设计结构

以逐点比较法为例子：



2.3 实际电路设计

设计原则按照：

1. 先时序后电路
2. 从原理结构到时序图，再到具体电路，形成完整的设计报告。

2.4 设计验证以及改进

首先，使用 MATLAB 等软件，实际仿真验证算法的正确性，得到算法的性能指标。改进算法，使之适合 FPGA 的操作，拥有更多的并行特性。

其次，利用 FPGA 的开发软件进行仿真，包括功能仿真和时序仿真。其中，功能仿真在布局布线之前，检查设计输入的正确性；时序仿真在布局布线之后，主要检查时序的收敛性，综合结果与功能仿真的不一致性。

最后，综合及布局布线和比特流生成，实现从 VHDL 语言到门级网表的生成。烧录到 FPGA 后，得到实际的性能指标。

3. 可能遇到的问题以及预期解决办法

1. 在比较积分法圆弧插补中，X、Y 两坐标轴的插补是同时进行的，但他们又是相互联系的，所以在实际设计中发现，如果按理论仅仅通过偏差判别，来分别发出 x、Y 方向的进给脉冲的话会因为某个轴的插补提前完成而进行不下去。

解决措施：设计为，如果某个轴插补结束，另外一个轴插补继续进行并发出进给脉冲。

2. 在利用比较积分法进行跨象限圆弧的插补时，会发现插补到最后误差会比较大，经过观察和分析发现较大的误差是由于每次过象限时产生的小误差所积累的。

解决措施：为解决大误差的问题，在 X、Y 轴插补电路中分别添加了一个判别，判断新的插补点到原点的距离是否超出了半径，超出的话即使偏差判别式为真也不输出进给脉冲。

3. 在数字积分法插补设计中，输出脉冲频率过低的问题

解决措施：需要进行左移规格化处理。

4. 在处理数字积分法圆弧终点判别时，采用两坐标分别控制的方式，会出现提前结束插补的情况。

解决措施：采用两坐标结合的终点判别方法。

5. 如何提高插补电路模块的最大允许频率，从而提高插补运算的速度

解决措施：对插补电路模块进行优化，得到更为简洁合理的插补电路

6. 插补算法直接用 VHDL 语言编程较为复杂，难以实现

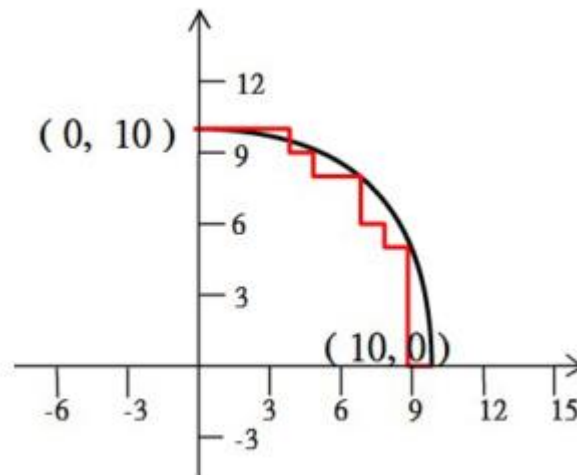
解决措施：将插补算法进一步转化为易于编程的递推形式

4. 最终成果预测

4.1 算法的软件仿真

计划首先使用 Matlab 建模，模拟实际的轨迹运行。电脑通过串行接口与 FPGA 通信，得到 FPGA 的控制信号。

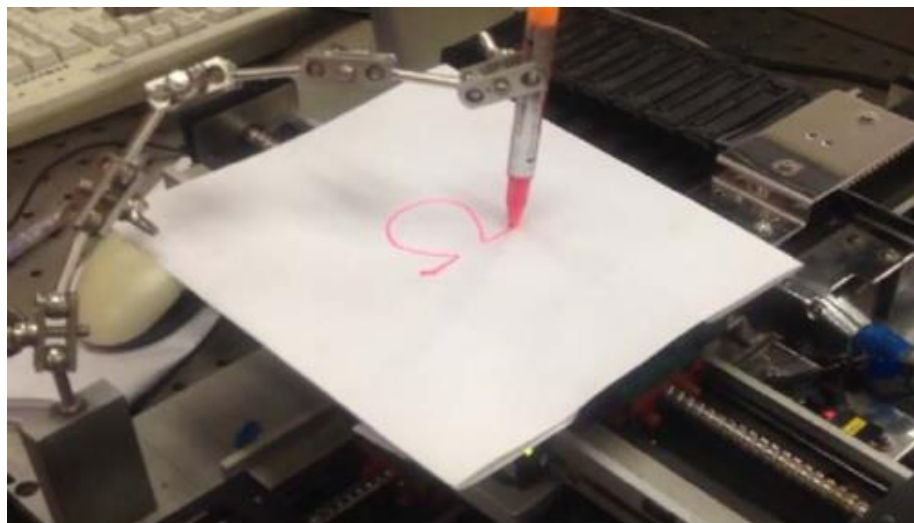
得到预期与实际路线的仿真结果，进一步通过 Matlab 分析算法的性能，为改进圆弧插补算法提供原始数据，为实际设计硬件电路提供指导。



预期软件仿真结果

4.2 实际硬件实现

预期在 FPGA 内部实现了三大经典脉冲增量算法：逐点比较法，数字积分法和比较积分法的逻辑电路设计，并通过了时序分析和时序仿真验证。对于数字积分法，考虑到它的实用性，在具体实现时，采用的是增加了左移规格化和余数寄存器预置数功能的改进电路，插补速度平稳性和精度方面得到了改善。



预期结果示意图

4.3 用户接口

外加键盘及 LCD 显示屏实现友好的用户接口。

最终，用户可以通过手动输入圆弧半径，起点和终点坐标，速度，加速度，旋转方向等信息，实现圆弧插补的匀速和匀加速控制。

在 LCD 屏上，开始显示功能选择菜单。当实际工作时，为用户提供输入信息的接口，包括：圆弧半径，起点和终点坐标，速度，加速度，旋转方向等信息。

显示当前的运行状态，包括速度，加速度，运行的坐标，预计时间等信息。



实际硬件示意图