注意到公式(7)的分母，知道这意味着什么吗？这就是 M 级流水的递归环路。实际上 M 倍超前计算就是将

$$H\left(e^{j\omega}\right)=\frac{b}{1-a\cdot z^{-1}}$$

分子分母同乘 $\sum_{i=0}^{M-1} a^{i}\cdot b\cdot z^{-i}$

$$H\left(e^{j\omega}\right)=\frac{\sum_{i=0}^{M-1} a^{i}\cdot b\cdot z^{-i}}{\left(1-a\cdot z^{-1}\right)\left(\sum_{i=0}^{M-1} a^{i}\cdot b\cdot z^{-i}\right)}$$

化简，得

$$H\left(e^{j\omega}\right)=\frac{\sum_{i=0}^{M-1} a^{i}\cdot b\cdot z^{-i}}{1-a^{M}\cdot z^{-M}}$$

图 1一阶 IIR 节的 M-1 步超前

对应的电路结构如下，

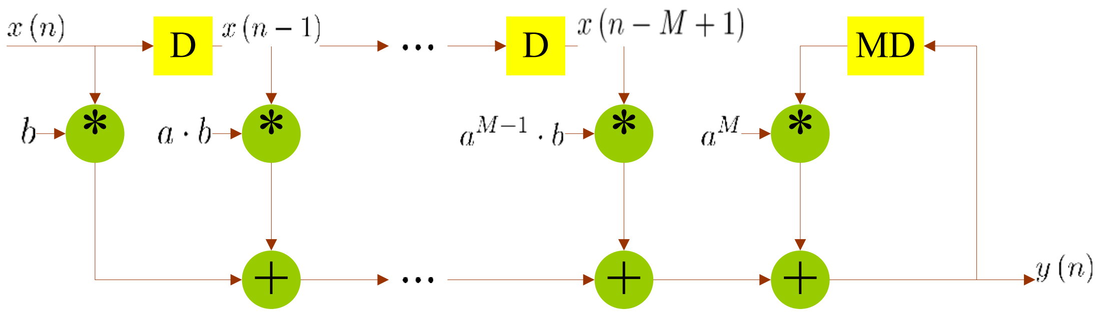


图 2一阶 IIR 节的 M-1 步超前结构

在递归环路中，出现了 M 个延时单元，这是由公式(7)分母所决定的，反过来，如果想得到一个 M 级流水的递归环路，就必须使得分母的最低次幂为z^{-M}（常数 1 除外）；此外应注意到分子部分变复杂了，原始迭代方程分子为 b，消耗一个乘法器，现在分子变为 $\sum_{i=0}^{M-1} a^{i}\cdot b\cdot z^{-i}$，消耗 M 个乘法器和 M-1 个加法器，M-1 个延时器，资源的增加是很可观的，在这一点上就输给 M 倍降速电路了。但是，但是，基于超前计算所得的 M 级递归结构，是不需要多个独立通道复合就能达到 100%的硬件利用率，从这点上说，基于超前计算的递归流水线技术仍有其用武之地。

对于一阶 IIR，分母从$1-a\cdot z^{-1}$变为$1-a^{M}\cdot z^{-M}$，意味着什么呢？从零极点图，能帮助我们看清楚其实真面目，

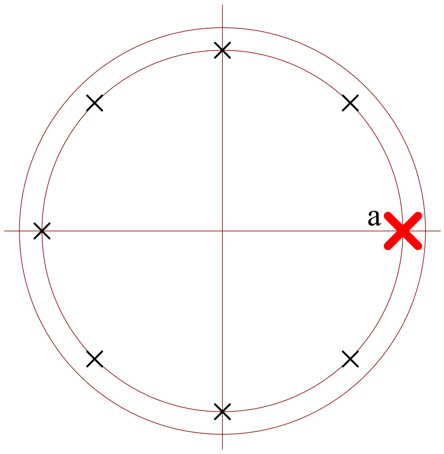


图 3分母为$1-a^8\cdot z^{-8}$的极点分布图

红色极点为原始极点，来自于$1-a\cdot z^{-1}$，黑色极点是新增极点，来自于$\sum_{i=0}^{M-1} a^i\cdot z^{-i}$。不知大家看出什么端倪没？其实，分母若为$1-a^M\cdot z^{-M}$，就表示圆 r=a 上等间隔的 M 个极点，反过来，想得到$1-a^M\cdot z^{-M}$形式的分母，就应使得圆 r=a 上出现等间隔的 M 个极点。对于图 10 所示的公式变换过程，实际上就是在 zp 平面添加“可相互抵消”的极点和零点，新添加的极点和原始极点刚好构成分母$1-a^M\cdot z^{-M}$,，这样就实现了 M 级递归流水线。

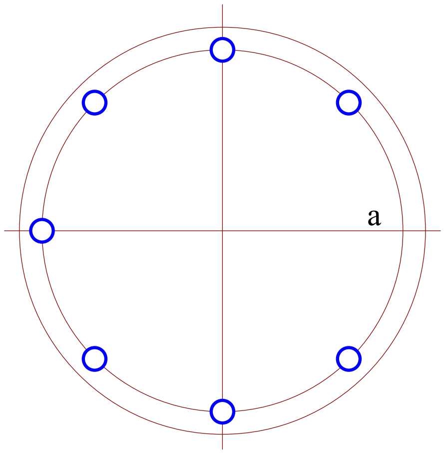


图 4分子$\sum_{i=0}^{M-1} a^i\cdot z^{-i}$所产生的零点

对应于新增的极点，产生相应的零点（用于抵消新增极点的影响），直接实现为$\sum_{i=0}^{M-1} a^i\cdot z^{-i}$，需要 M 个乘法器，M-1 个加法器和延时单元。直接实现还有一个坏处就是噪声敏感度高，一般采用级联实现，将以上零点恰当的组合就能获得节约面积的结构，一般选择正多边形分解法,，比如图 13 就可以做如下分解，

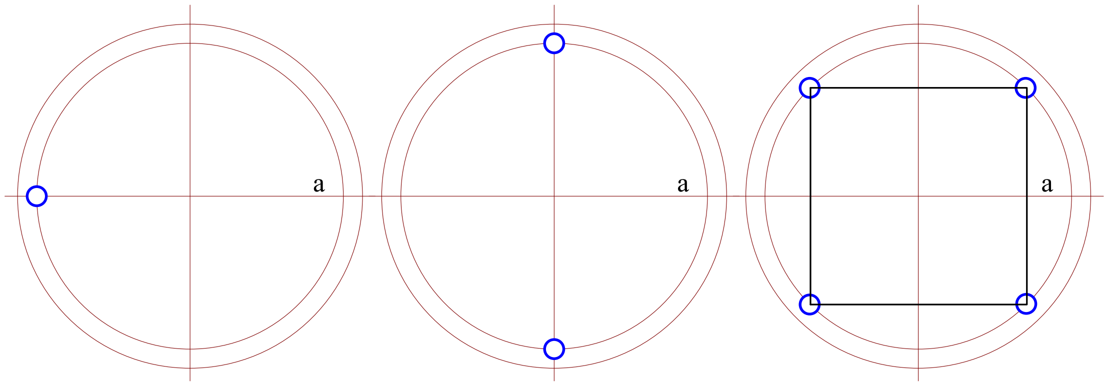


图 5分子$\sum_{i=0}^{M-1} a^i \cdot z^{-i}$的级联实现，最大可能节约资源

分解之后的系统框图为，

$$\boxed{1+a\cdot z^{-1}} \rightarrow \boxed{1+a^2\cdot z^{-2}} \rightarrow \boxed{1+a^4\cdot z^{-4}} \rightarrow \boxed{\frac{1}{1-a^8\cdot z^{-8}}} \rightarrow$$

图 6分解实现框图

显然，非递归部分的乘法器资源和加法器资源得到了一定程度的节约。下面在来练习一个分解的例子，分子为$\sum_{i=0}^{16-1} a^i \cdot z^{-i}$的零点分布如下图

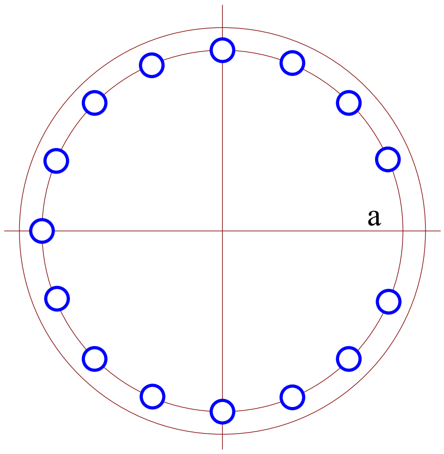


图 7$\sum_{i=0}^{16-1} a^i \cdot z^{-i}$的零点分布图

分解图为

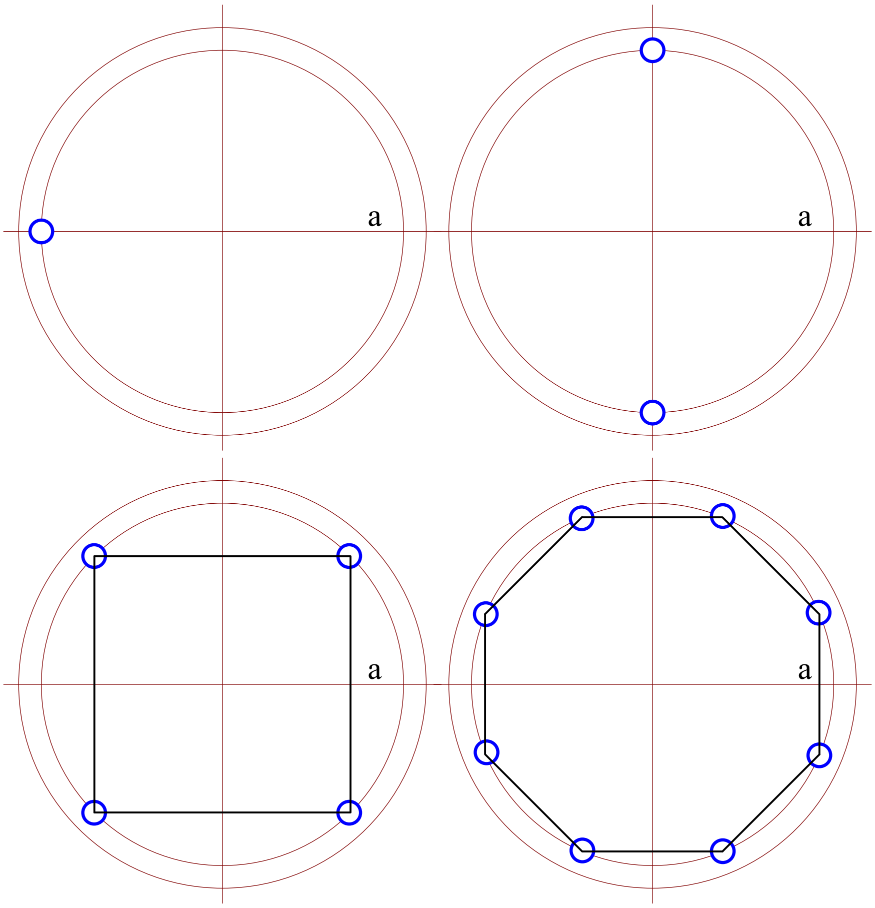


图 8

分解后的系统框图如下

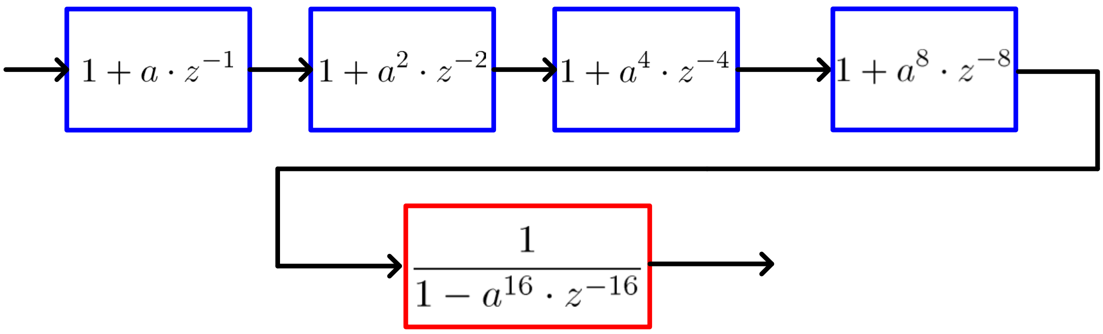


图 9

将以上过程，编制为 Maple 程序，以便于自动求解，

代码 1　求解 M 级流水的一阶 IIR 节

```
> restart :
>
> M := 8;
                                  M := 8
>
> Dz := 1-1/2·p;
```

$$Dz := 1 - \frac{1}{2}\ p$$

> $Pz := 1 + sum(q_i \cdot p^i,\ i = 1..M-1);$

$$Pz := 1 + q_1\ p + q_2\ p^2 + q_3\ p^3 + q_4\ p^4 + q_5\ p^5 + q_6\ p^6 + q_7\ p^7$$

>

> $DP := collect(Dz \cdot Pz,\ p);$

$$DP := 1 - \frac{1}{2}\ q_7\ p^8 + \left(q_7 - \frac{1}{2}\ q_6\right)\ p^7 + \left(q_6 - \frac{1}{2}\ q_5\right)\ p^6 + \left(q_5 - \frac{1}{2}\ q_4\right)\ p^5 + \left(-\frac{1}{2}\ q_3 + q_4\right)\ p^4 + \left(q_3 - \frac{1}{2}\ q_2\right)\ p^3 + \left(q_2 - \frac{1}{2}\ q_1\right)\ p^2 + \left(q_1 - \frac{1}{2}\right)\ p$$

>

> $eqns := \{seq(coeff(DP,\ p,\ i) = 0,\ i = 1..M-1)\};$

$$eqns := \left\{q_1 - \frac{1}{2} = 0,\ q_2 - \frac{1}{2}\ q_1 = 0,\ q_3 - \frac{1}{2}\ q_2 = 0,\ -\frac{1}{2}\ q_3 + q_4 = 0,\ q_5 - \frac{1}{2}\ q_4 = 0,\ q_6 - \frac{1}{2}\ q_5 = 0,\ q_7 - \frac{1}{2}\ q_6 = 0\right\}$$

> $vars := [seq(q_i,\ i = 1..M-1)];$

$$vars := [q_1,\ q_2,\ q_3,\ q_4,\ q_5,\ q_6,\ q_7]$$

> $sols := solve(eqns,\ vars);$

$$sols := \left[\left[q_1 = \frac{1}{2},\ q_2 = \frac{1}{4},\ q_3 = \frac{1}{8},\ q_4 = \frac{1}{16},\ q_5 = \frac{1}{32},\ q_6 = \frac{1}{64},\ q_7 = \frac{1}{128}\right]\right]$$

>

> $subs(op(sols),\ [op(eqns)]);$

$$[0 = 0,\ 0 = 0,\ 0 = 0,\ 0 = 0,\ 0 = 0,\ 0 = 0,\ 0 = 0]$$

>

> $DP := subs(op(sols),\ DP);$

$$DP := 1 - \frac{1}{256}\ p^8$$

> $Nz := factor(subs(op(sols),\ Pz));$

$$Nz := \frac{1}{128}\ (p + 2)\ \left(p^2 + 4\right)\ \left(p^4 + 16\right)$$

>

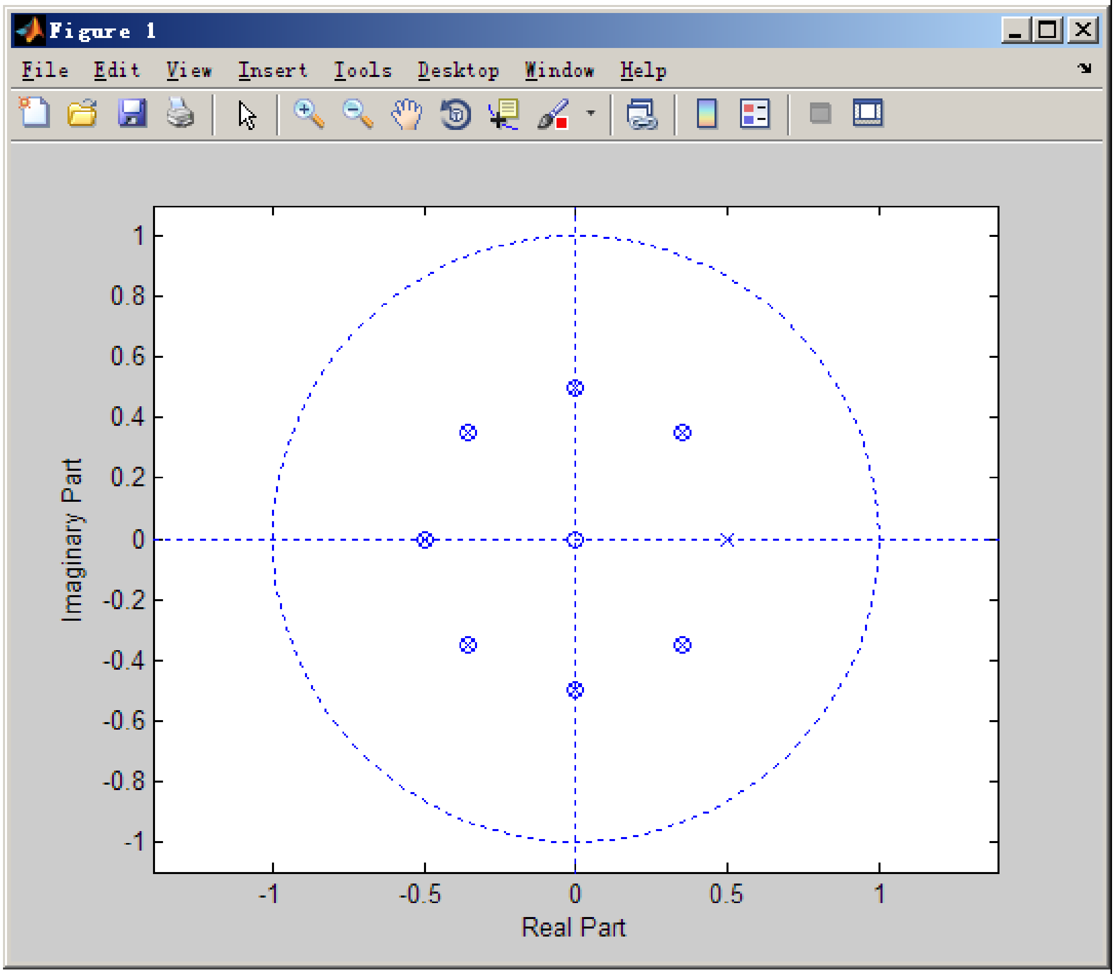
```
> maplematrix_b := Matrix([seq(coeff(Nz, p, i), i = 0..M
      - 1)]);
```

$$maplematrix\_b := \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{4} & \frac{1}{8} & \frac{1}{16} & \frac{1}{32} & \frac{1}{64} & \frac{1}{128} \end{bmatrix}$$

```
> > maplematrix_a := Matrix([seq(coeff(DP, p, i), i = 0..M)]);
```

$$maplematrix\_a := \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{1}{256} \end{bmatrix}$$

```
>
> Matlab[setvar]("matlabmatrix_b", maplematrix_b);
> Matlab[setvar]("matlabmatrix_a", maplematrix_a);
> Matlab[evalM]("zplane(matlabmatrix_b,matlabmatrix_a);");
```



利用数学归纳法，容易证明，具有 M 级流水递归环路的 IIR 一阶环初始值为，

$$y\left(i-M\right)=a^{-M+i+1}\cdot y\left(-1\right) \tag{1}$$

其中，$i=0,1,...,M-2$。

有时，流水级数 M 并非 2 的次幂，那么分子的零点分布就不是那么“美妙”，分解起来就没那么顺利。使用代码 1，设计一个 12 级流水的 IIR 一阶节，设计结果如下（MAPLE），

$$H\left(e^{j\omega}\right)=\frac{\left(z^{-1}+2\right)\left(z^{-2}+4\right)\left(z^{-2}+2z^{-1}+4\right)\left(z^{-2}-2z^{-1}+4\right)\left(z^{-4}-4z^{-2}+16\right)/2048}{1-z^{-12}/4096}$$

(2)

公式(9)分子是MAPLE的默认因式分解的结果，零极点图如下

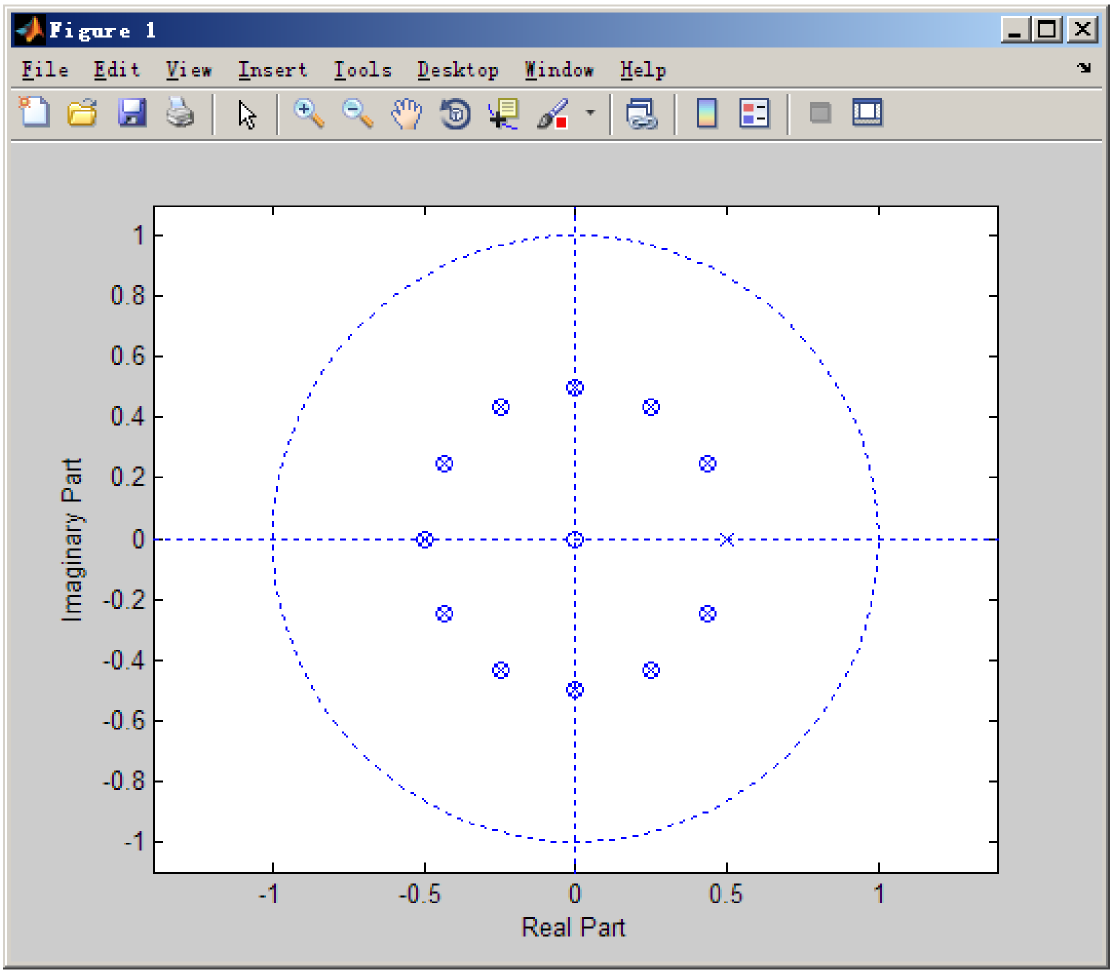
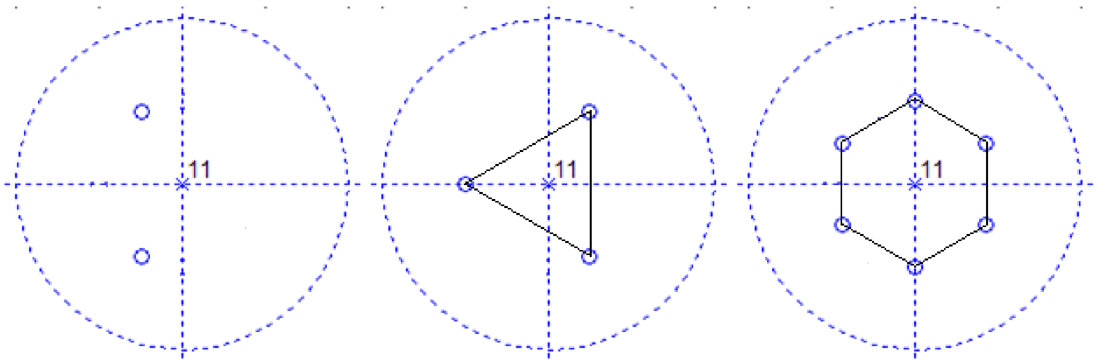


图 10　12 级流水环路的 IIR 一阶节零极图

直接观察公式(9)也能得出一种分解，即将分子的右边三个因子乘开，得

$$\frac{1}{2048}(p+2)\left(p^2+4\right)\left(p^8+16p^4+256\right)$$

不过，从系数的噪声敏感性来说，总不希望得到较高阶次的环，比如$\left(p^8+16p^4+256\right)$。另外，也可以使用正多边形分解原则来分解，见下图，



对应的分子为，

$$\frac{1}{2048}\left(p^2+2p+4\right)\left(p^3+8\right)\left(p^6+64\right)$$

关于一阶IIR节就介绍那么多，实际应用往往是二阶IIR节，或更高阶！区别于一阶IIR节，二阶IIR节的超前计算环路可能会出现不稳定的情况。以公式(10)为例

$$H\left(e^{j\omega}\right)=\frac{1}{1+a_1z^{-1}+a_2z^{-2}} \tag{3}$$

构造M级流水环路，就是要将z^{-1}项的阶数提高到z^{-M}。一般分两种做法，各有其优缺点，请大家课后思考！

1. 聚类超前流水线

所谓的聚类超前流水线，就是分子分母同乘$1+\sum_{i=1}^{M-1} q_i\cdot z^{-i}$，然后令分母中阶数在$-1,\ldots,-M+1$之间的项的系数为0，从而求出待定系数$q_i$。以公式(11)传递函数为例，编写MAPLE求解程序如下，

$$H\left(e^{j\omega}\right)=\frac{1}{1-\frac{5}{4}z^{-1}+\frac{3}{8}z^{-2}} \tag{4}$$

代码 2　求解 M 级聚类超前流水的二阶 IIR 节

```
> restart:
> M := 3;
                    M := 3
> Dz := 1-5/4·p + 3/8·p^2;
                    Dz := 1 - 5/4 p + 3/8 p^2
> Pz := 1 + sum(q[i]·p^i, i = 1..M-1);
                    Pz := 1 + q1 p + q2 p^2
>
> DP := collect(Dz·Pz, p);
```

$$DP := 1 + \frac{3}{8}\, q_2\, p^4 + \left(-\frac{5}{4}\, q_2 + \frac{3}{8}\, q_1\right) p^3 + \left(q_2 - \frac{5}{4}\, q_1 + \frac{3}{8}\right) p^2 + \left(q_1 - \frac{5}{4}\right) p$$

\>

\> eqns := {seq(coeff(DP, p, i) = 0, i = 1 .. M − 1)};

$$eqns := \left\{q_1 - \frac{5}{4} = 0,\ q_2 - \frac{5}{4}\, q_1 + \frac{3}{8} = 0\right\}$$

\> vars := [seq(q_i, i = 1 .. M − 1)];

$$vars := [q_1,\ q_2]$$

\> sols := solve(eqns, vars);

$$sols := \left[\left[q_1 = \frac{5}{4},\ q_2 = \frac{19}{16}\right]\right]$$

\>

\> subs(op(sols), [op(eqns)]);

$$[0 = 0,\ 0 = 0]$$

\>

\>

\> DP := subs(op(sols), DP);

$$DP := 1 + \frac{57}{128}\, p^4 - \frac{65}{64}\, p^3$$

\> Nz := factor(subs(op(sols), Pz));

$$Nz := 1 + \frac{5}{4}\, p + \frac{19}{16}\, p^2$$

\> maplematrix_b := Matrix([seq(coeff(Nz, p, i), i = 0 .. M − 1)]);

$$maplematrix\_b := \begin{bmatrix} 1 & \frac{5}{4} & \frac{19}{16} \end{bmatrix}$$

\> maplematrix_a := Matrix([seq(coeff(DP, p, i), i = 0 .. M + 1)]);

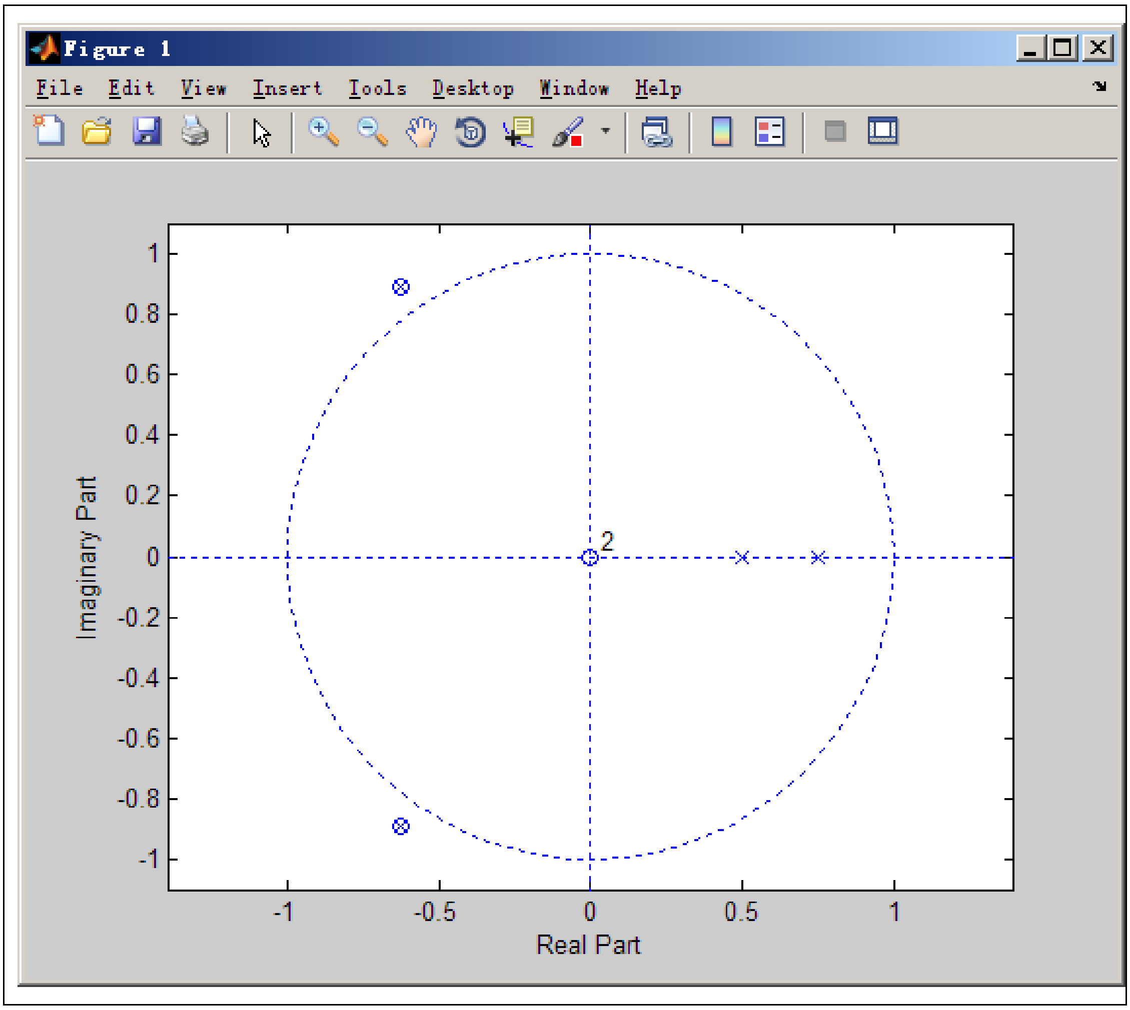
$$maplematrix\_a := \begin{bmatrix} 1 & 0 & 0 & -\frac{65}{64} & \frac{57}{128} \end{bmatrix}$$

\>

\> Matlab[setvar]("matlabmatrix_b", maplematrix_b);

\> Matlab[setvar]("matlabmatrix_a", maplematrix_a);

\> Matlab[evalM]("zplane(matlabmatrix_b,matlabmatrix_a);");



从零极点图可知，代码2的3级聚类流水设计并不稳定，2个极点落到单位圆之外。课本上给出一个研究结论，即存在一个临界的值M_c，当$M \geqslant M_c$时设计总是稳定的，M_c可用公式进行估计（请查阅相关文献）。为了简便，我们可以逐渐增加M值直到设计稳定，对以上例子，当M=5时得到稳定设计，有

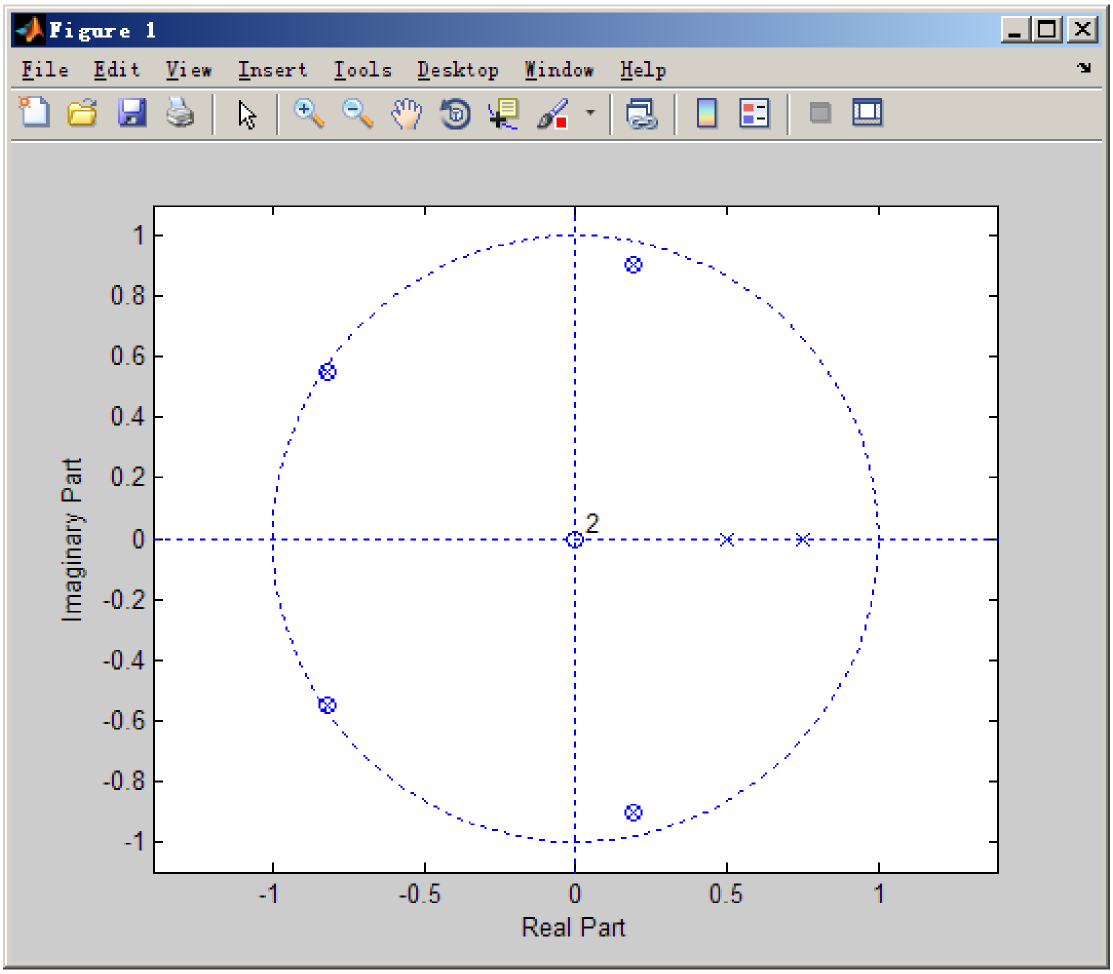


图 11 稳定的聚类流水线设计，对应公式(11)，取 M=5

其实M=5，虽能得到稳定设计，但是极点离单位圆非常近；一般M再大一些会得到“更稳定”的设计，比如M=8时，有

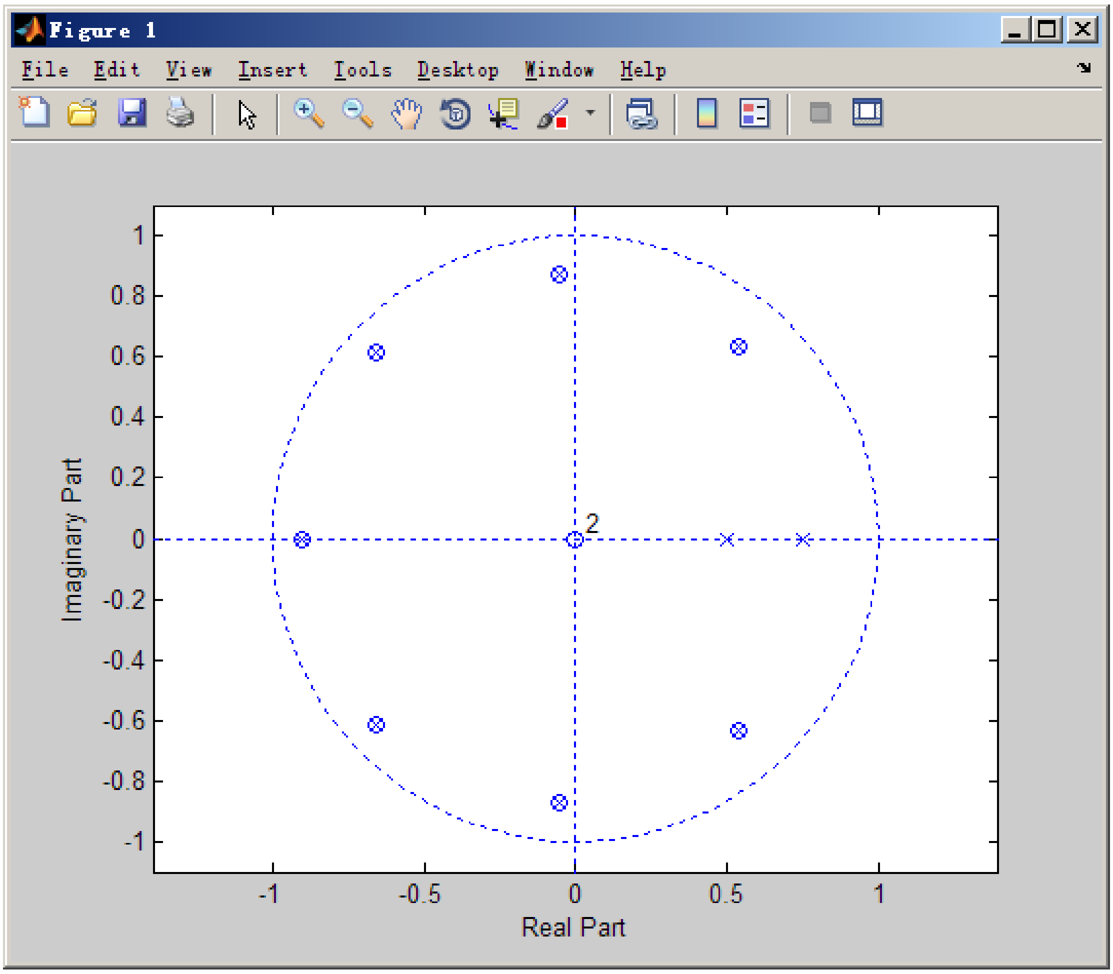


图 12　稳定的聚类流水线设计，对应公式(11)，取 M=8

有时，我们只需要3级流水环路，而不是5级以上的流水环路，是不是就不能用聚类流水线了呢？这里给出一个“更为先进的”设计方法，

代码 3　先进的 2 阶 IIR 节聚类超前流水线设计

```
> restart :
>
>
> M := 3; N := 1;
                    M := 3
                    N := 1
>
>
> Dz := 1 - 5/4·p + 3/8·p^2;
```

$$Dz := 1 - \frac{5}{4}\,p + \frac{3}{8}\,p^2$$

```
> Pz := 1 + sum(q[i]·p^i, i = 1..M − 1 + N);
```

$$Pz := 1 + q_1\, p + q_2\, p^2 + q_3\, p^3$$

```
>
> DP := collect(Dz·Pz, p);
```

$$DP := 1 + \frac{3}{8}\, q_3\, p^5 + \left(-\frac{5}{4}\, q_3 + \frac{3}{8}\, q_2\right) p^4 + \left(q_3 - \frac{5}{4}\, q_2 + \frac{3}{8}\, q_1\right) p^3 + \left(q_2 - \frac{5}{4}\, q_1 + \frac{3}{8}\right) p^2 + \left(q_1 - \frac{5}{4}\right) p$$

```
>
> eqns := {seq(coeff(DP, p, i) = 0, i = 1..M + N)};
```

$$eqns := \left\{q_1 - \frac{5}{4} = 0,\ -\frac{5}{4}\, q_3 + \frac{3}{8}\, q_2 = 0,\ q_2 - \frac{5}{4}\, q_1 + \frac{3}{8} = 0,\ q_3 - \frac{5}{4}\, q_2 + \frac{3}{8}\, q_1 = 0\right\}$$

```
> eqns := eqns\{seq(coeff(DP, p, i) = 0, i = M)};
```

$$eqns := \left\{q_1 - \frac{5}{4} = 0,\ -\frac{5}{4}\, q_3 + \frac{3}{8}\, q_2 = 0,\ q_2 - \frac{5}{4}\, q_1 + \frac{3}{8} = 0\right\}$$

```
>
>
> vars := [seq(q[i], i = 1..M − 1 + N)];
```

$$vars := [q_1,\ q_2,\ q_3]$$

```
> sols := solve(eqns, vars);
```

$$sols := \left[\left[q_1 = \frac{5}{4},\ q_2 = \frac{19}{16},\ q_3 = \frac{57}{160}\right]\right]$$

```
>
> subs(op(sols), [op(eqns)]);
```

$$[0 = 0,\ 0 = 0,\ 0 = 0]$$

```
>
>
> DP := subs(op(sols), DP);
```

$$DP := 1 + \frac{171}{1280}\, p^5 - \frac{211}{320}\, p^3$$

```
> Nz := factor(subs(op(sols), Pz));
```

$$Nz := 1 + \frac{5}{4}\, p + \frac{19}{16}\, p^2 + \frac{57}{160}\, p^3$$

```
> maplematrix_b := Matrix([seq(coeff(Nz, p, i), i = 0..M
    − 1)]);
```

$$maplematrix\_b := \begin{bmatrix} 1 & \frac{5}{4} & \frac{19}{16} \end{bmatrix}$$

```
> maplematrix_a := Matrix([seq(coeff(DP, p, i), i=0..M
     +1)]);
```

$$maplematrix\_a := \begin{bmatrix} 1 & 0 & 0 & -\frac{211}{320} & 0 \end{bmatrix}$$

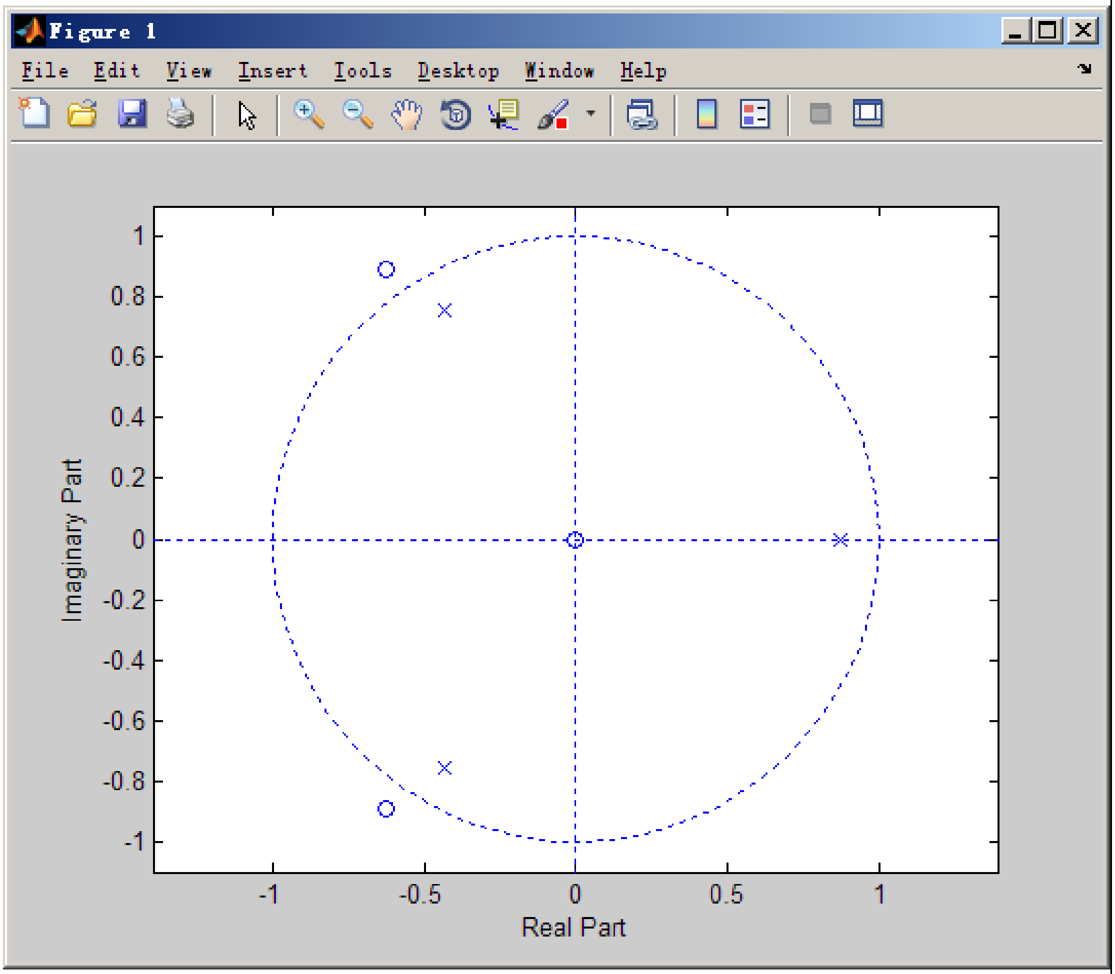
```
>
> Matlab[setvar]("matlabmatrix_b", maplematrix_b);
> Matlab[setvar]("matlabmatrix_a", maplematrix_a);
> Matlab[evalM]("zplane(matlabmatrix_b,matlabmatrix_a);");
>
> evalf(maplematrix_a);
```

$$\begin{bmatrix} 1. & 0. & 0. & -0.6593750000 & 0. \end{bmatrix}$$

```
>
> evalf(map(abs, [solve(DP)]));
```

[2., 1.333333333, 2.348151569, 1.093350732, 1.093350732]

```
>
```



不见得每个2阶IIR节都能这么设计，很多时候得借助搜索算法进行优化才能找到稳定解。实

际编程根据问题的不同而不同，但先进聚类流水线的思想掌握以后，就能灵活对付各种不同问题。在这个代码3的例子中，分母分子同乘的多项式比一般聚类多一阶（也可以多2阶或更高阶），然后指定约束是分母中尽可能少的保留分母中阶数大于等于M的项，这里保留了两项（3阶项和5阶项），很幸运的是，设计出来的滤波器正好是稳定的。“关于使用优化算法的高效设计，大家可以根据这个思路去自行编写求解程序”。

2. 离散超前流水线

稍稍区别于聚类超前流水线，离散超前流水线设计总是稳定（当然，前提是原滤波器是稳定的），但是离散超前可能会耗费更多的资源（包括乘法器，加法器和延时）。

所谓的离散，就是将公式(10)变为公式(12)的形式，注意观察分母的变化，

$$H\left(e^{j\omega}\right)=\frac{P\left(z\right)}{\left(1+a_1z^{-1}+a_2z^{-2}\right)P\left(z\right)}=\frac{P\left(z\right)}{1+\bar{a}_1z^{-M}+\bar{a}_2z^{-2M}} \tag{5}$$

原来的z^{-1}变为z^{-M}，原来的z^{-2}变为z^{-2M}；退而广之，原来的z^{-i}变为z^{-iM}，其中 i 为正整数。分母的各项之间阶数间隔为 M，这就是离散的含义。

仍然以公式(11)为例，设计 3 级离散超前流水线，代码如下，

代码 4　2 阶 IIR 节离散超前流水线设计

```
> restart :
>
> M := 3;
                                M := 3
> Dz := 1 - 5/4·p + 3/8·p^2;
                        Dz := 1 - 5/4 p + 3/8 p^2
> k := degree(Dz, p);
                                k := 2
> K := M·k;
                                K := 6
> Pz := 1 + sum(a[i]·p^i, i = 1 .. K - k) :
>
> DP := collect(Dz·Pz, p) :
>
> eqns := {seq(coeff(DP, p, i) = 0, i = 1 .. K)} :
> eqns := eqns\{seq(coeff(DP, p, i·M) = 0, i = 1 .. k)} :
>
> vars := [seq(a[i], i = 1 .. K - k)] :
> sols := solve(eqns, vars) :
>
> subs(op(sols), [op(eqns)]);
```

$$[0 = 0,\ 0 = 0,\ 0 = 0,\ 0 = 0]$$

>

>

> $DP := subs(op(sols), DP);$

$$DP := 1 + \frac{27}{512}\ p^6 - \frac{35}{64}\ p^3$$

> $Nz := factor(subs(op(sols), Pz));$

$$Nz := \frac{1}{64}\ \left(9\ p^2 + 12\ p + 16\right)\ \left(p^2 + 2\ p + 4\right)$$

> $maplematrix\_b := Matrix([seq(coeff(Nz, p, i), i = 0..K - k)]);$

$$maplematrix\_b := \begin{bmatrix} 1 & \frac{5}{4} & \frac{19}{16} & \frac{15}{32} & \frac{9}{64} \end{bmatrix}$$

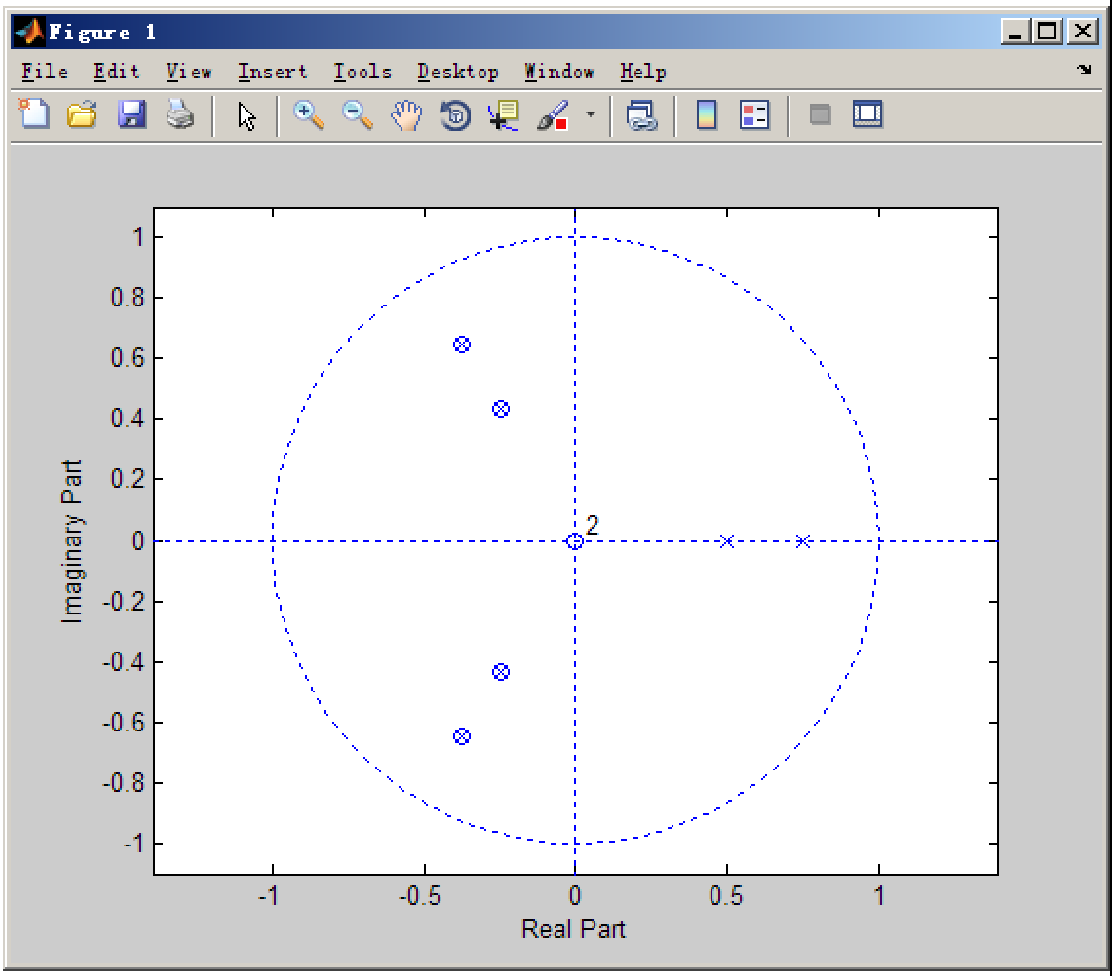
> $maplematrix\_a := Matrix([seq(coeff(DP, p, i), i = 0..K)]);$

$$maplematrix\_a := \begin{bmatrix} 1 & 0 & 0 & -\frac{35}{64} & 0 & 0 & \frac{27}{512} \end{bmatrix}$$

> Matlab[setvar]("matlabmatrix_b", maplematrix_b);

> Matlab[setvar]("matlabmatrix_a", maplematrix_a);

> Matlab[evalM]("zplane(matlabmatrix_b,matlabmatrix_a);");



从离散超前流水线的零极点图可以看出，对于原始的每一个级点，都会增加“一圈相互抵消的零极点”。而且不论 M 取何值，如果原 2 阶 IIR 节稳定，新设计的离散超前流水 IIR 节必然稳定。