La transformée en Z

Pourquoi ?

Outil important pour résoudre et raisonner sur les équations récurrentes linéaires stationnaires,

Permet d'abord de comprendre des outils de conception tels que Simulink, Scicos, voire Lustre-Scade

- Définition
- Equations récurrentes
- Fonctions puissance
- Exemples
- Théorèmes des valeurs initiales et finales
- Signaux et systèmes
- Systèmes rationnels en Z
- Z, Laplace et Euler
- Stabilité en Z
- Programmation

Définition

A toute suite x, on associe

$$\mathcal{Z}x(z) = \sum_{0}^{\infty} x(n) z^{-n}$$

(sous réserve d'existence)

Pourquoi ? ? ?

Equations récurrentes

Parce que cela permet de transformer des équations récurrentes en équations algébriques ordinaires sur lesquelles on peut « calculer normalement »

grâce à deux propriétés :

1. Linéarité :

$$\mathcal{Z}(\alpha x + \beta y) = \alpha \mathcal{Z} x + \beta \mathcal{Z} y$$

2. Les décalages d'indices sont transformés en produits :

   Soit $x^{+}(n) = x(n+1)$

   On a :

$$\mathcal{Z}(x^{+})(z) = z(\mathcal{Z}x(z) - x(0))$$

Démonstration

$$\sum_{0}^{\infty} x^{+}(n) z^{-n} = \sum_{0}^{\infty} x(n+1) z^{-n}$$

$$\sum_{0}^{\infty} x^{+}(n) z^{-n} = z \sum_{0}^{\infty} x(n+1) z^{-(n+1)}$$

$$\sum_{0}^{\infty} x^{+}(n) z^{-n} = z \sum_{1}^{\infty} x(n) z^{-n}$$

$$\sum_{0}^{\infty} x^{+}(n) z^{-n} = z \left(\sum_{0}^{\infty} x(n) z^{-n} - x(0) \right)$$

Exemple

Equation récurrente de premier ordre, linéaire, à coefficients constants

$$y^+ = ay + bx$$

$$\mathcal{Z}(y^+)(z) = z(\mathcal{Z}y(z) - y(0)) = a\mathcal{Z}y(z) + b\mathcal{Z}x(z)$$

$$(z-a)\mathcal{Z}y(z) = b\mathcal{Z}x(z) + zy(0)$$

$$\boxed{\mathcal{Z}y(z) = \frac{1}{z-a}(b\mathcal{Z}x(z) + zy(0))}$$

L'équation récurrente a été « résolue » et la solution est une fraction rationnelle

Fonctions puissance

Les fonctions puissance

Posons

$$G_k(n) = C_n^k a^{n-k}$$

où on admet que

$$n < k \Rightarrow C_n^k = 0$$

On a alors

$$\boxed{\mathcal{Z}G_k(z) = \frac{z}{(z-a)^{k+1}}}$$

Démonstration

Par induction : $k = 0$

$$\sum_{0}^{\infty} a^n z^{-n} = \frac{1}{1 - \frac{a}{z}} = \frac{z}{z - a}$$

pourvu que $|\frac{a}{z}| < 1$

Démonstration

$k+1$

On veut calculer

$$\sum_{k+1}^{\infty} C_n^{k+1} a^{n-k-1} z^{-n}$$

On remarque que

$$\frac{\partial}{\partial a} a^{n-k} = (n-k) a^{n-k-1}$$

et que

$$C_n^{k+1} = \frac{n-k}{k+1} C_n^k$$

Donc

$$\sum_{k+1}^{\infty} C_n^{k+1} a^{n-k-1} z^{-n} = \frac{1}{k+1} \frac{\partial}{\partial a} \sum_{k}^{\infty} C_n^k a^{n-k} z^{-n}$$

Démonstration

$k+1$

Donc

$$\sum_{k+1}^{\infty} C_n^{k+1} a^{n-k-1} z^{-n} = \frac{1}{k+1} \frac{\partial}{\partial a} \sum_{k}^{\infty} C_n^k a^{n-k} z^{-n}$$

On applique l'hypothèse de récurrence

$$\sum_{k+1}^{\infty} C_n^{k+1} a^{n-k-1} z^{-n} = \frac{1}{k+1} \frac{\partial}{\partial a} \frac{z}{(z-a)^{k+1}}$$

Or

$$\frac{\partial}{\partial a} \frac{z}{(z-a)^{k+1}} = (k+1) \frac{z}{(z-a)^{k+2}}$$

Donc

$$\sum_{0}^{\infty} C_n^{k+1} a^{n-k-1} z^{-n} = \frac{z}{(z-a)^{k+2}}$$

Exemples de signaux

Un échelon : $u(n) = 1$

C'est le cas $k = 0, a = 1$

$$\mathcal{Z}u(z) = \frac{z}{z-1}$$

Une rampe : $r(n) = n$ C'est le cas $k = 1, a = 1$

$$\mathcal{Z}r(z) = \frac{z}{(z-1)^2}$$

Exemple de systèmes

Fibonnacci :

$$f(n+2) = f(n+1) + f(n), \qquad f(0) = f(1) = 1$$

On peut le réecrire :

$$f^{++}(n) = f^{+}(n) + f(n), \qquad f(0) = f(1) = 1$$

$$\mathcal{Z}f^{++}(z) = z(\mathcal{Z}f^{+}(z) - 1)$$

$$\mathcal{Z}f^{+}(z) = z(\mathcal{Z}f(z) - 1)$$

$$z(z(\mathcal{Z}f(z) - 1) - 1) = z(\mathcal{Z}f(z) - 1) + \mathcal{Z}f(z)$$

Exemple de systèmes

Fibonnacci :

$$z(z(\mathcal{Z}f(z)-1)-1) = z(\mathcal{Z}f(z)-1) + \mathcal{Z}f(z)$$

On « résoud »

$$z^2\mathcal{Z}f(z) - z^2 - z = z\mathcal{Z}f(z) - z + \mathcal{Z}f(z)$$

$$(z^2 - z - 1)\mathcal{Z}f(z) = z^2$$

$$\mathcal{Z}f(z) = \frac{z^2}{z^2 - z - 1}$$

Résolution

Chercher les pôles (racines du dénominateur)

solution de l'équation du 2ème degré

$$z^2 - z - 1 = 0$$

$$z_0 = \frac{1+\sqrt{5}}{2} \qquad z_1 = \frac{1-\sqrt{5}}{2}$$

C'est le fameux nombre d'or aux propriétés magiques

Résolution

Décomposition des fractions rationnelles en éléments simples :

$$\frac{z^2}{(z-z_0)(z-z_1)} = \frac{zA}{z-z_0} + \frac{zB}{z-z_1}$$

On multiplie par $z - z_0$

$$\frac{z(z-z_0)}{(z-z_0)(z-z_1)} = \frac{A(z-z_0)}{z-z_0} + \frac{B(z-z_0)}{z-z_1}$$

$$\frac{z}{(z-z_1)} = A + \frac{B(z-z_0)}{z-z_1}$$

On fait $z = z_0$

$$\frac{z_0}{(z_0-z_1)} = A$$

Résolution

Décomposition des fractions rationnelles en éléments simples :

$$\frac{z^2}{(z-z_0)(z-z_1)} = \frac{zA}{z-z_0} + \frac{zB}{z-z_1}$$

On multiplie par $z - z_1$

$$\frac{z(z-z_1)}{(z-z_0)(z-z_1)} = \frac{A(z-z_1)}{z-z_0} + \frac{B(z-z_1)}{z-z_1}$$

$$\frac{z}{(z-z_0)} = \frac{A(z-z_1)}{z-z_0} + B$$

On fait $z = z_1$

$$\frac{z_1}{(z_1-z_0)} = B$$

Résolution

$$\mathcal{Z}f(z) = \frac{z^2}{z^2 - z - 1}$$

$$z_0 = \frac{1+\sqrt{5}}{2} \qquad z_1 = \frac{1-\sqrt{5}}{2}$$

$$\mathcal{Z}f(z) = \frac{z_0}{z_0 - z_1}\frac{z}{z - z_0} + \frac{z_1}{z_1 - z_0}\frac{z}{z - z_1}$$

$$f(n) = \frac{1}{\sqrt{5}}\left[\left(\frac{1+\sqrt{5}}{2}\right)^{n+1} - \left(\frac{1-\sqrt{5}}{2}\right)^{n+1}\right]$$

Que c'est beau ! ! !

Autres propriétés

Théorème de la valeur initiale :

$$\boxed{x(0) = \lim_{z \longrightarrow \infty} \mathcal{Z}x(z)}$$

si les limites existent

Théorème de la valeur finale :

$$\boxed{\lim_{n \longrightarrow \infty} x(n) = \lim_{z \longrightarrow 1} (z-1)\mathcal{Z}x(z)}$$

si les limites existent

Démonstration

Théorème de la valeur initial :

$$\mathcal{Z}x(z) = x(0) + \sum_{1,\infty} x(n)z^{-n}$$

Quand z tend vers ∞, $z^{-n}, n > 1$ tend vers 0

Démonstration

Théorème de la valeur initiale :

$$(z-1)\mathcal{Z}x(z) = \sum_{0,\infty} x(n)z^{-(n-1)} - \sum_{0,\infty} x(n)z^{-n}$$

$$(z-1)\mathcal{Z}x(z) = \sum_{-1,\infty} x(n+1)z^{-n} - \sum_{0,\infty} x(n)z^{-n}$$

$$\sum_{-1,N} x(n+1)z^{-n} - \sum_{0,N} x(n)z^{-n} = x(N+1)$$

Applications

Comment connaître la valeur finale de la réponse à un échelon d'un système de premier ordre

$$\mathcal{Z}y(z) = \frac{bz}{(z-a)}\mathcal{Z}x(z)$$

avec

$$\mathcal{Z}x(z) = \frac{z}{(z-1)}$$

sans calculer la solution ?

Par le théorème de la valeur finale, il suffit de faire :

$$\lim_{z\to 1}(z-1)\frac{bz}{z-a}\frac{z}{z-1} = \frac{b}{1-a}$$

Systèmes et signaux, convolution

Question : quelle différence entre signaux et systèmes (linéaires stationnaires) ?

presque aucune !

exemple : système du premier ordre

$$\mathcal{Z}y(z) = \frac{z}{z-a}\mathcal{Z}x(z)$$

$$\mathcal{Z}y(z) = \mathcal{Z}S(z)\mathcal{Z}x(z)$$

Avec

$$\mathcal{Z}S(z) = \frac{z}{z-a}$$

Tous deux sont représentés par des transformées en Z :

- Comment expliquer cela ?
- Quelle est la vision signal d'un système ?

Convolution discrète

Convolution de deux signaux $x,\ y$:

$$(x\ *\ y)(n) = \sum_{0}^{\infty} x(m)y(n-m)$$

Théorème de convolution

$$\boxed{\mathcal{Z}(x*y)(z) = \mathcal{Z}x(z)\mathcal{Z}y(z)}$$

A tout signal discret correspond un système dont la sortie s'obtient en convolant l'entrée avec le signal.

Démonstration

$$\mathcal{Z}(x*y)(z) = \sum_{n=0,\infty} \left(\sum_{m=0,\infty} x(m)y(n-m) \right) z^{-n}$$

$$\mathcal{Z}(x*y)(z) = \sum_{n=0,\infty} \sum_{m=0,\infty} x(m)y(n-m)z^{-m-(n-m)}$$

Posons $u = n - m$

$$\mathcal{Z}(x*y)(z) = \sum_{m=0,\infty} \sum_{u=0,\infty} x(m)y(u)z^{-m}z^{-u}$$

$$\mathcal{Z}(x*y)(z) = \sum_{m=0,\infty} x(m)z^{-m} \sum_{u=0,\infty} y(u)z^{-u}$$

$$\mathcal{Z}(x*y)(z) = \mathcal{Z}x(z)\mathcal{Z}y(z)$$

Quel est le signal associé à un système ?

$$\mathcal{Z}S(z) = \frac{z}{z-a}$$

On a

$$\mathcal{Z}Y(z) = \mathcal{Z}S(z)\mathcal{Z}X(z)$$

$\mathcal{Z}S(z)$ est la transformée de la réponse du système au signal d'entrée δ de transformée $\mathcal{Z}\delta(z) = 1$

$\mathcal{Z}S(z)$ est la fonction de transfert du système

$\mathcal{Z}^{-1}\mathcal{Z}S(z)$ est réponse impulsionnelle du système

La sortie du système (linéaire stationnaire) s'obtient en convolant l'entrée avec la réponse impulsionnelle du système

Exemples

Opérateur unité

$$\mathcal{Z}\delta(z) = \sum_{0}^{\infty} \delta(n) z^{-n} = 1$$

$$\delta(n) = \begin{cases} 1 & \text{si } n = 0 \\ 0 & \text{sinon} \end{cases}$$

Exemples

Intégration discrète

$$
\begin{aligned}
y(n) &= \sum_{m=0,n} x(n-m) \\
&= \sum_{m=0,n} s(m)x(n-m)
\end{aligned}
$$

D'où

$$s(n) = 1$$

L'intégration discrète s'obtient en convolant avec l'échelon unité

La fonction de transfert correspondante est

$$\mathcal{Z}u(z) = \frac{z}{z-1}$$

Diagrammes de blocs

Représentation graphique très populaire :

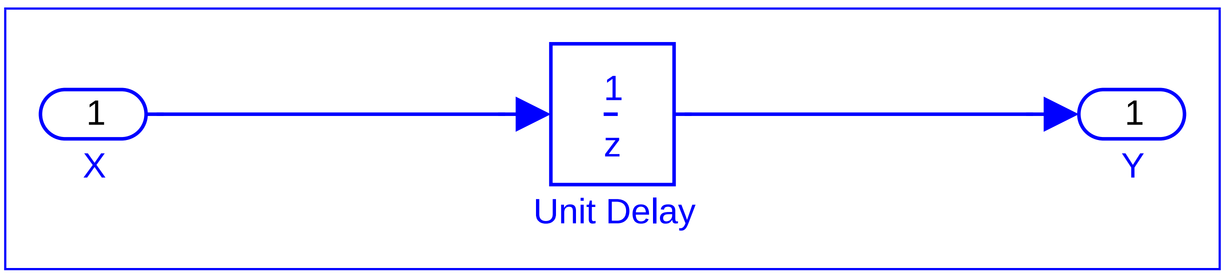
Réseau de boîtes et de fils :

Les boîtes sont des opérateurs (systèmes élémentaires) et des systèmes

Les fils sont les signaux qui sont transformés par les boîtes.

Exemple :

$$y^+ = x$$

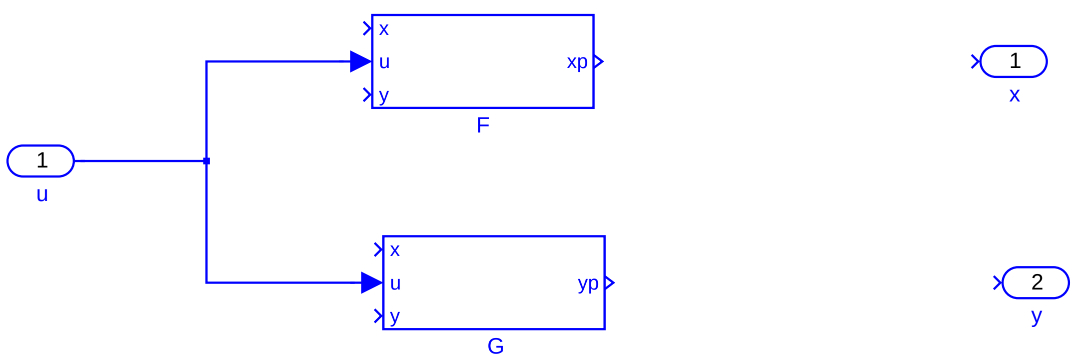


Des systèmes d'équations aux diagrammes

Un système

$$
\begin{aligned}
x^+ &= f(x, y, u) \\
y^+ &= g(x, y, u)
\end{aligned}
$$

Construire f, g

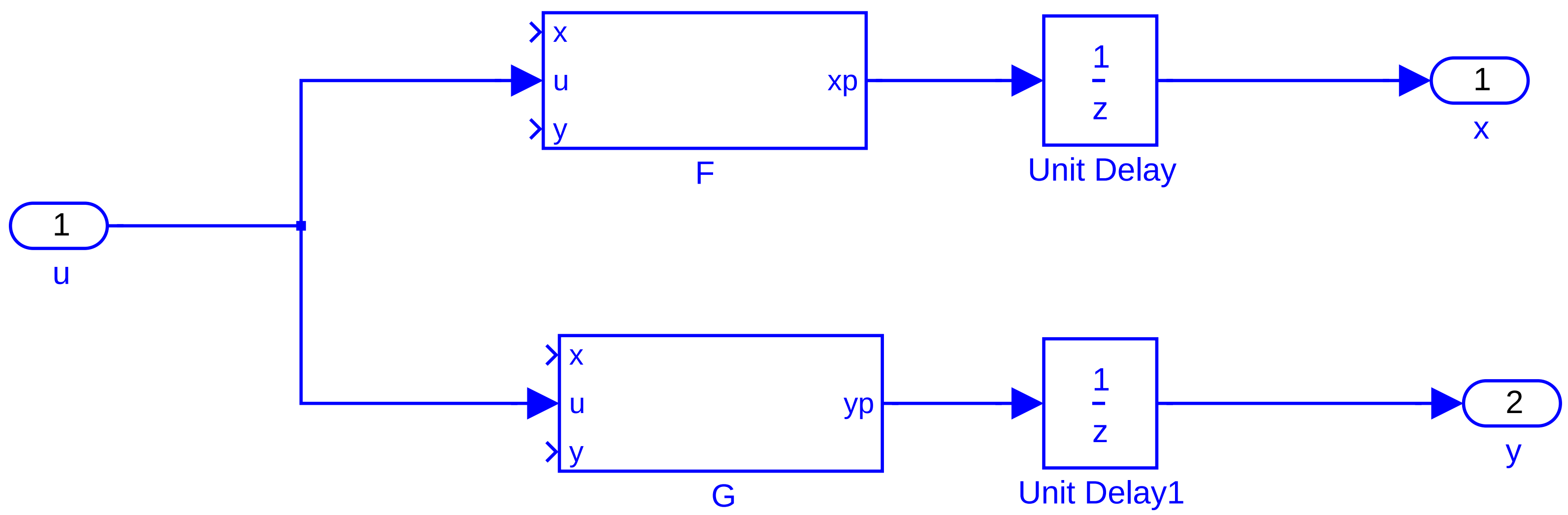


Des systèmes d'équations aux diagrammes

Un système

$$
\begin{aligned}
x^+ &= f(x, y, u) \\
y^+ &= g(x, y, u)
\end{aligned}
$$

Retarder

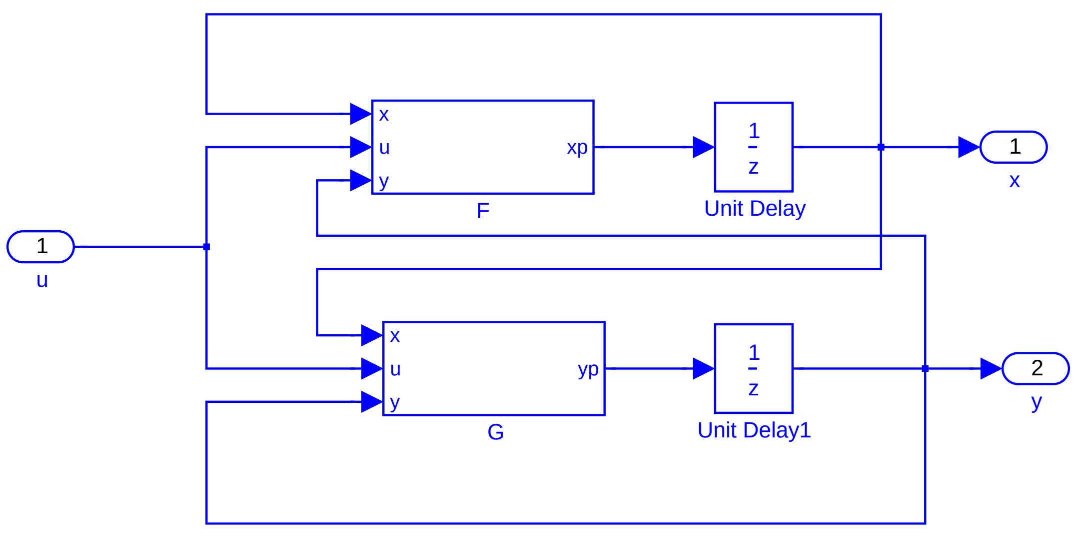


Des systèmes d'équations aux diagrammes

Un système

$$
\begin{aligned}
x^+ &= f(x, y, u) \\
y^+ &= g(x, y, u)
\end{aligned}
$$

Reboucler



Systèmes rationnels

Forme générale : $\dfrac{N(z)}{D(z)}$ avec degré N $\leq$degré D.

$$Y(z) = \frac{\sum_0^n a_i z^i}{\sum_0^{n-1} b_i z^i + z^n} X(z)$$

$$(\sum_0^{n-1} b_i z^i + z^n) Y(z) = \sum_0^n a_i z^i X(z)$$

$$z^n Y(z) = \sum_0^n a_i z^i X(z) - \sum_0^{n-1} b_i z^i Y(z)$$

$$Y(z) = \sum_0^n a_i z^{i-n} X(z) - \sum_0^{n-1} b_i z^{i-n} Y(z)$$

$$Y(z) = a_n X(z) + \sum_0^{n-1} z^{i-n} (a_i X(z) - b_i Y(z))$$

Systèmes rationnels

$$Y(z) = a_n X(z) + \sum_{0}^{n-1} z^{i-n}(a_i X(z) - b_i Y(z))$$

Posons :

$$\begin{aligned}
U_0(z) &= z^{-1}(a_0 X(z) - b_0 Y(z)) \\
\ldots & \\
U_{i+1}(z) &= z^{-1}(a_{i+1} X(z) - b_{i+1} Y(z) + U_i(z)) \\
\ldots & \\
Y(z) &= a_n X(z) + U_n(z)
\end{aligned}$$

Il est facile de vérifier algébriquement que ces deux expressions sont égales.

Systèmes rationnels

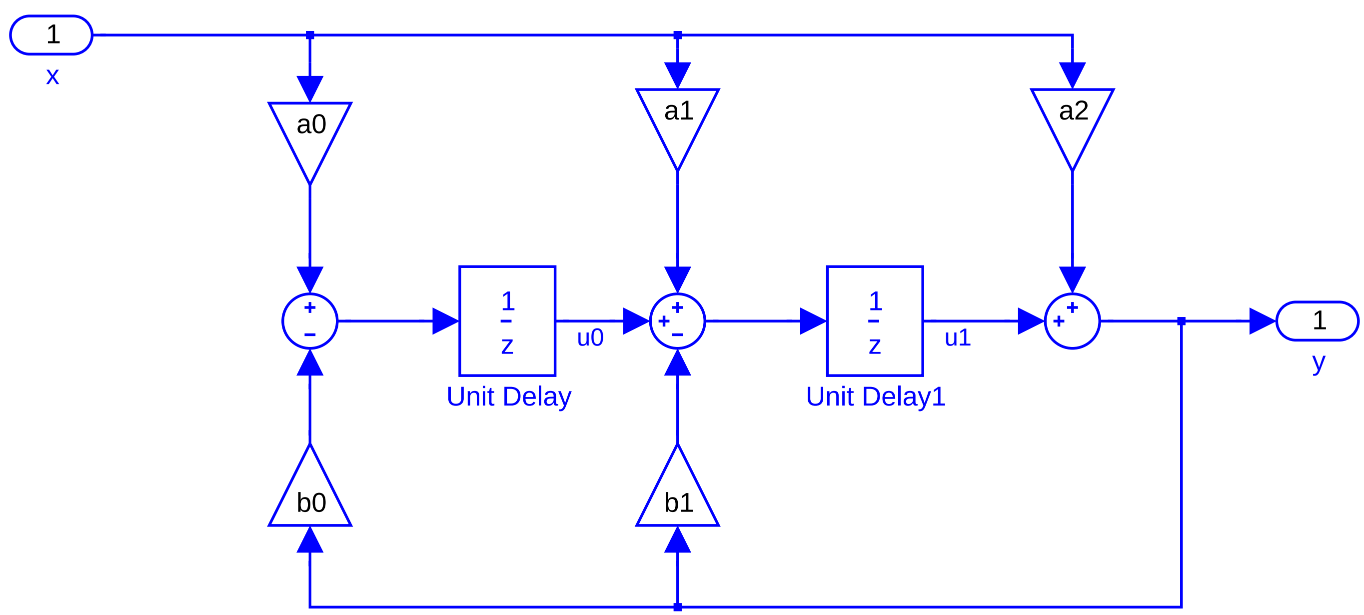
$$
\begin{aligned}
U_0(z) &= z^{-1}(a_0 X(z) - b_0 Y(z)) \\
\dots & \\
U_{i+1}(z) &= z^{-1}(a_{i+1} X(z) - b_{i+1} Y(z) + U_i(z)) \\
\dots & \\
Y(z) &= a^n X(z) + U_n(z)
\end{aligned}
$$

Conclusion : pour simuler un système rationnel d'ordre n, il suffit d'utiliser n retards.

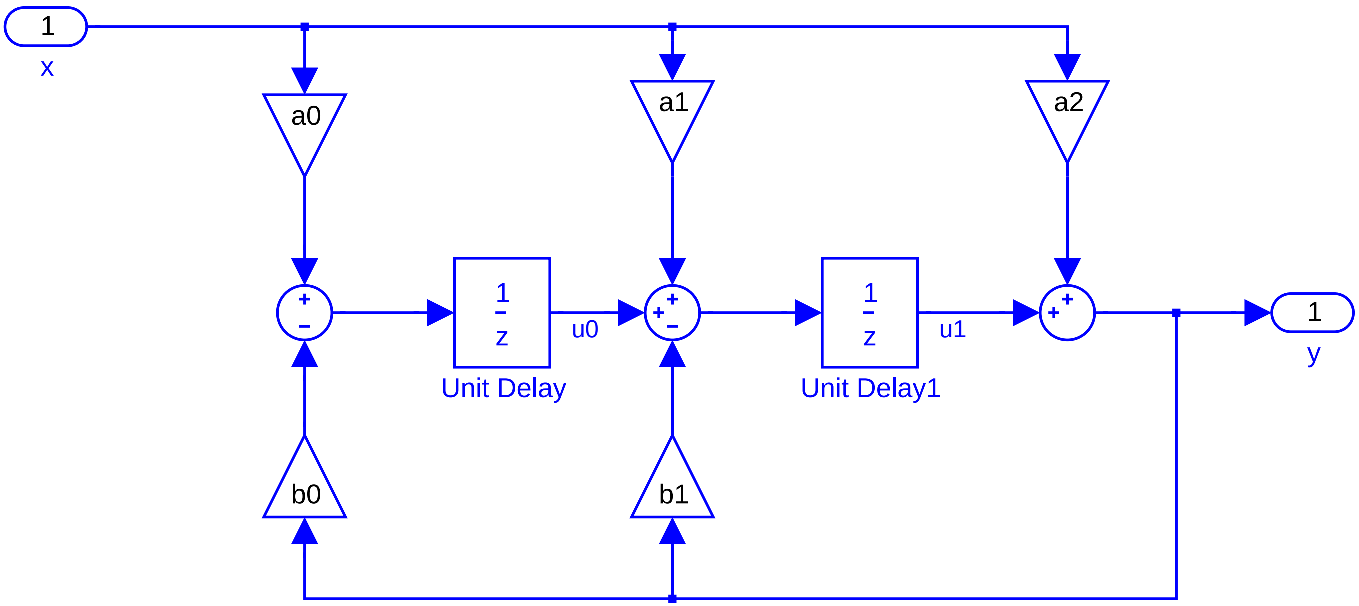
Systèmes rationnels

Exemple :

$$\frac{a_2 z^2 + a_1 z + a_0}{z^2 + b_1 z + b_0}$$



Systèmes rationnels



$$
\begin{aligned}
U_0(z) &= z^{-1}(a_0X(z) - b_0Y(z)) \\
U_1(z) &= z^{-1}(a_1X(z) - b_1Y(z) + U_0(z)) \\
Y(z) &= a_2X(z) + U_1(z) \\
Y(z) &= a_2X(z) + z^{-1}(a_1X(z) - b_1Y(z) + z^{-1}(a_0X(z) - b_0Y(z))) \\
Y(z) &= a_2X(z) + z^{-1}a_1X(z) + z^{-2}a_0X(z) - z^{-1}b_1Y(z) - z^{-2}b_0Y(z) \\
z^2Y(z) &= a_2z^2X(z) + a_1zX(z) + a_0X(z) - b_1zY(z) - b_0Y(z)
\end{aligned}
$$

Z, Laplace,...

$\frac{1}{z}$ est l'opérateur retard unité

Si on choisit un pas d'échantillonnage T on peut donc écrire :

$$\frac{1}{z} = e^{-sT}$$

On a alors l'approximation du premier ordre (développement limité)

$$e^{-sT} \approx 1 - sT$$

Donc, au premier ordre,

$$\begin{aligned} 1 - sT &\approx \frac{1}{z} \\ sT &\approx 1 - \frac{1}{z} = \frac{z-1}{z} \\ s &\approx \frac{z-1}{zT} \end{aligned}$$

Z, Laplace, et Euler

$$\boxed{s \approx \frac{z-1}{zT}}$$

Et Euler ?

Appliquons à une fonction de transfert H :

$$H(s) \approx H(\frac{z-1}{zT})$$

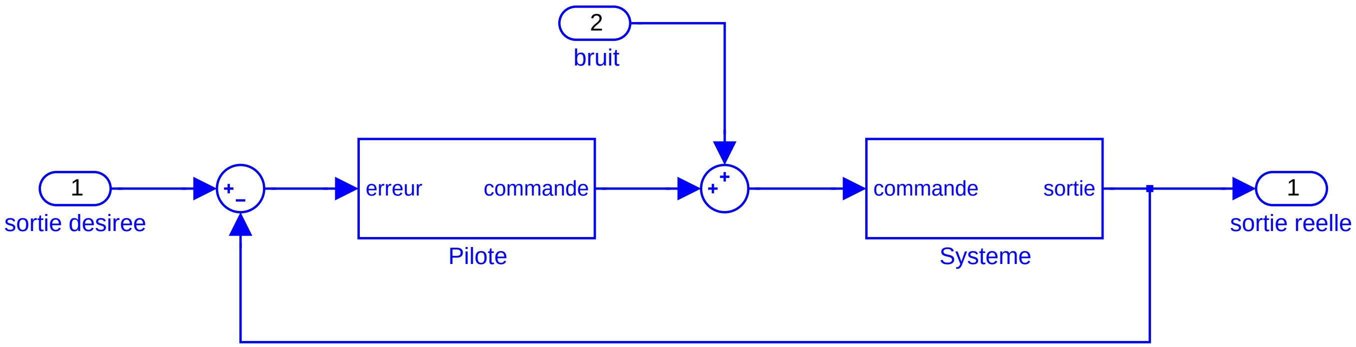
On est passé d'un système continu à un système échantillonné, implantable sur calculateur

Cela correspond à la méthode d'Euler :

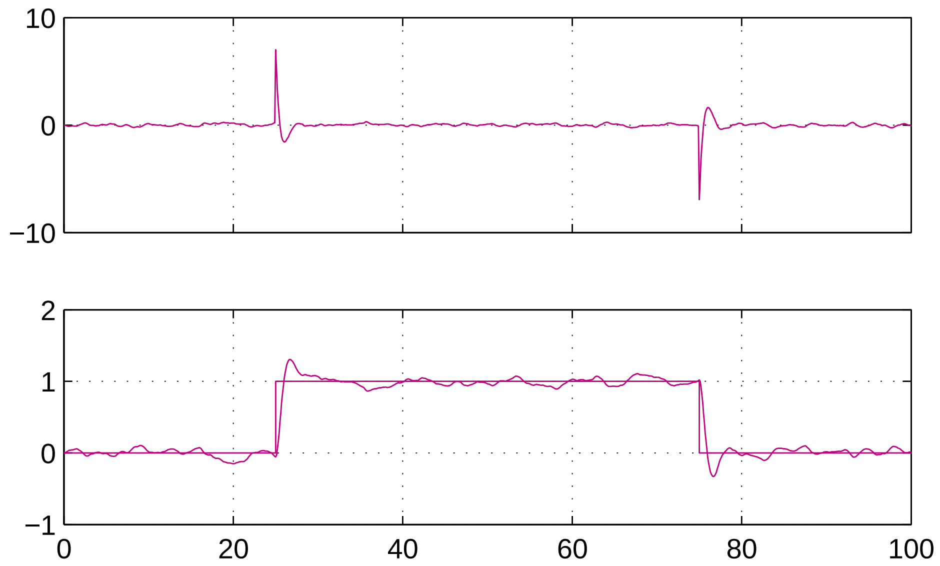
$$sx(s) \approx \frac{z-1}{zT} x(z)$$

$$x'(t) \approx \frac{x(n) - x(n-1)}{T}$$

Application



$$P = \frac{6.8s + 4}{s + 3.8} \qquad S = \frac{1}{s^2}$$



Time offset: 0

Application

Choix de la période :

On prend $T = \dfrac{0.1}{3.8} \approx 0.025$

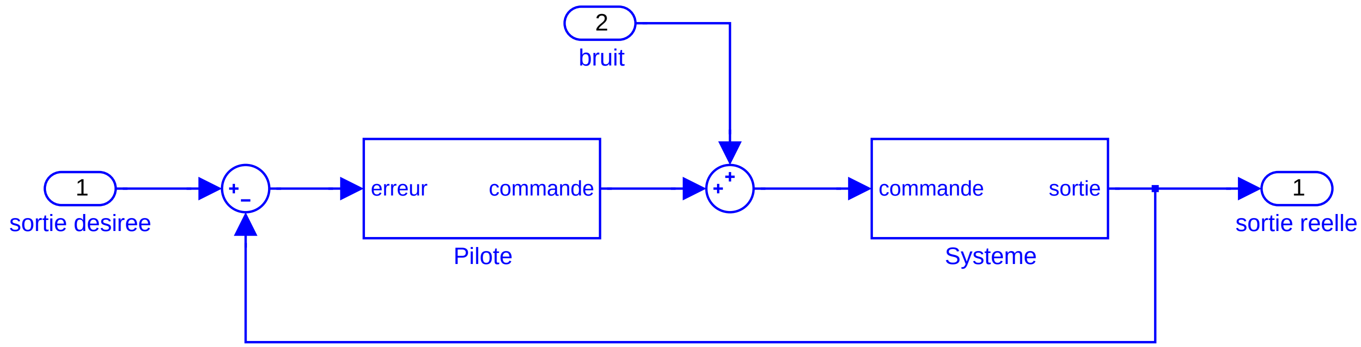
Transformation du pilote

$$P' = \frac{6.8\frac{z-1}{zT} + 4}{\frac{z-1}{zT} + 3.8}$$

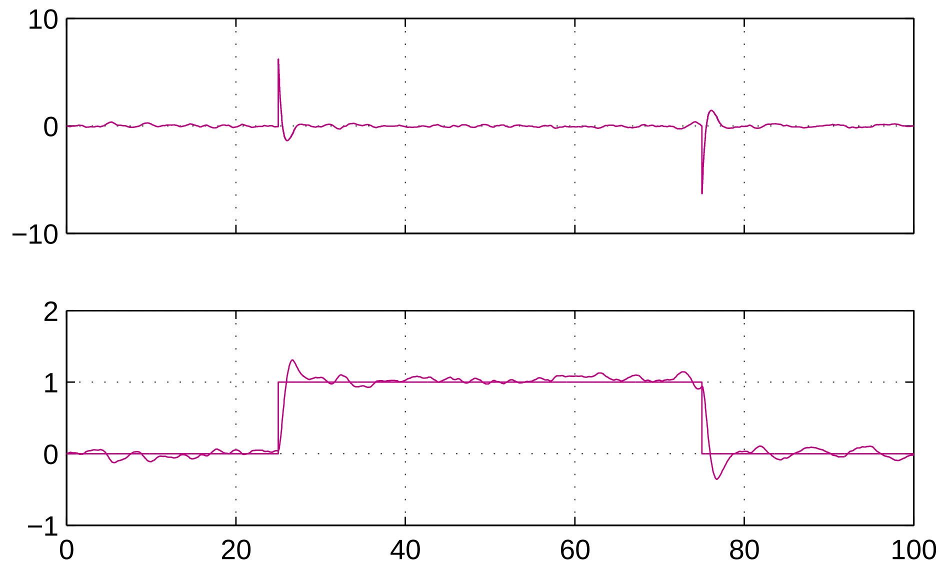
$$P' = \frac{6.9z - 6.8}{1.1z - 1}$$

On essaie

Application

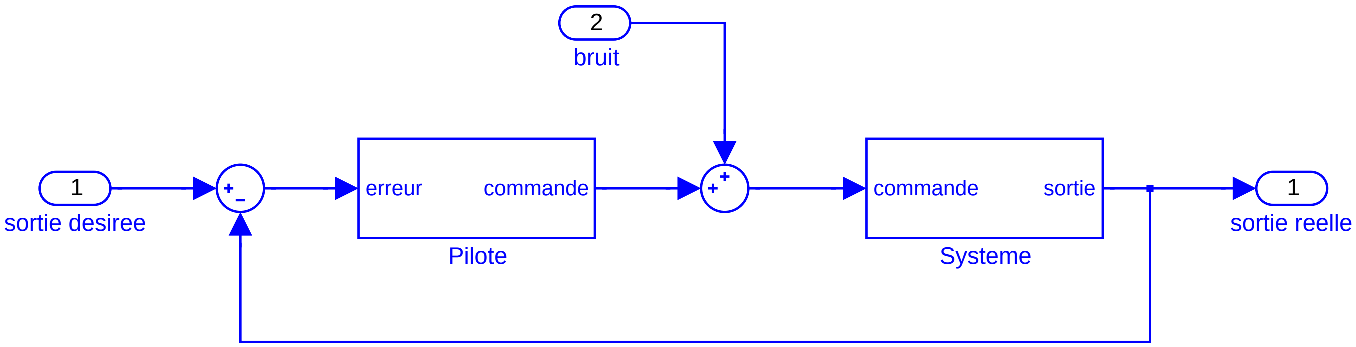


$$P = \frac{6.9z - 6.8}{1.1z - 1} \qquad S = \frac{1}{s^2}$$



Time offset: 0

Application



$$P = \frac{6.9z - 6.8}{1.1z - 1} \qquad S = \frac{1}{s^2}$$

On a donc su :

- calculer un pilote continu donnant de bons résultats
- le transformer en pilote numérique qui donne d'aussi bons résultats

Stabilité en Z

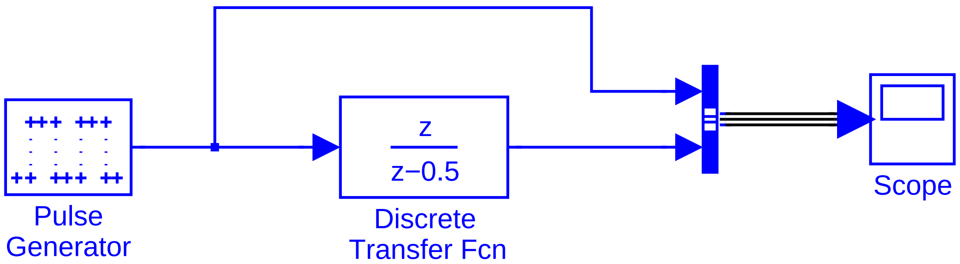
Un sujet très important, une propriété globale

- Stabilité
- Systèmes rationnels

Stabilité

Un système déplacé d'un point d'équilibre reconverge-t-il vers celui-ci ou diverge-t-il ?

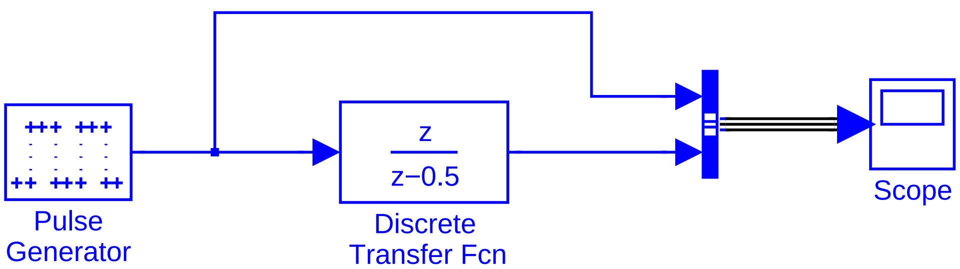
Exemple : Un système stable

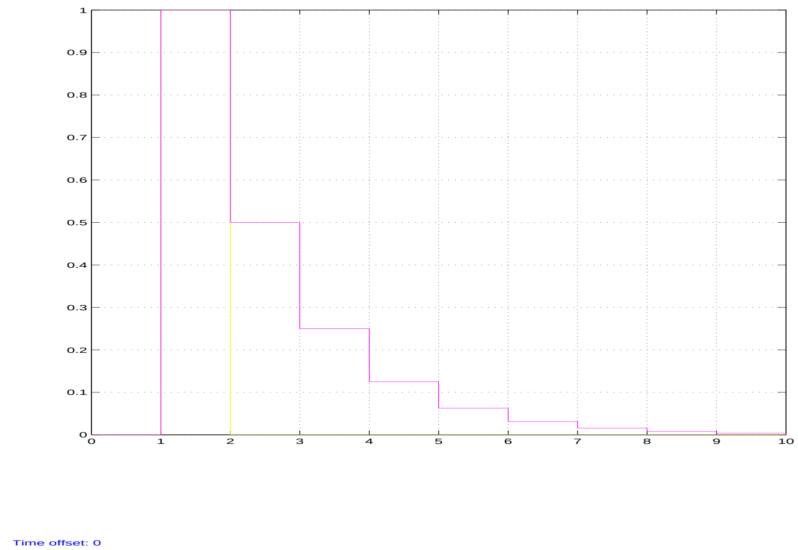


Stabilité

Un système déplacé d'un point d'équilibre reconverge-t-il vers celui-ci ou diverge-t-il ?

Exemple : Un système stable

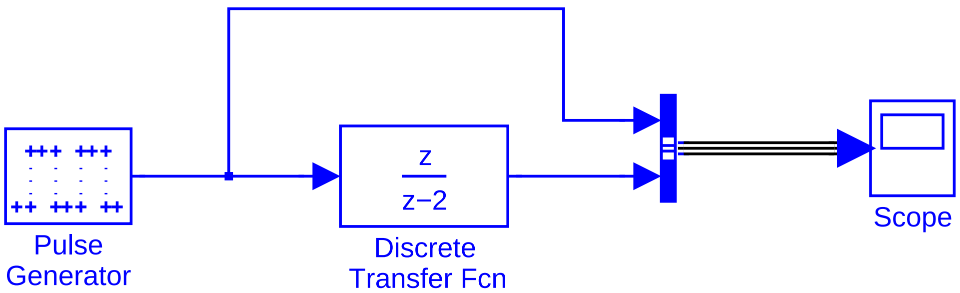




Stabilité

Un système déplacé d'un point d'équilibre reconverge-t-il vers celui-ci ou diverge-t-il ?

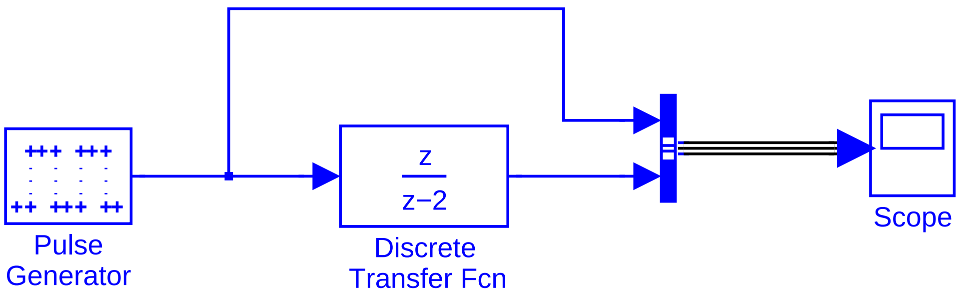
Exemple : Un système instable

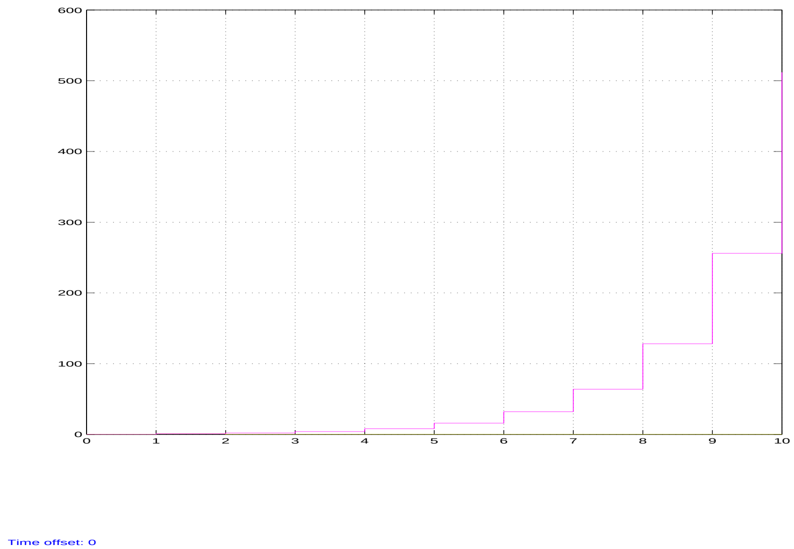


Stabilité

Un système déplacé d'un point d'équilibre reconverge-t-il vers celui-ci ou diverge-t-il ?

Exemple : Un système instable





Pourquoi ?

Systèmes rationnels

Il suffit de regarder les pôles (racines du dénominateur) de la fonction de transfert dans le plan complexe :

Si les pôles sont tous de module inférieur à 1 le système est dit asymptotiquement stable

en effet la réponse impulsionnelle sera une somme de termes de type

$$C_n^k a^{n-k}$$

avec a en module inférieur à 1. Tous ces termes tendent vers 0 quand n tend vers l'infini.

Exemple : $\dfrac{z}{z - 0.5}$

il y a un seul pôle de valeur absolue $0.5 < 1$: le système est asymptotiquement stable

Programmation

Les schémas-bloc discrets (échantillonnés) s'appliquent aussi aux opérations non linéaires (tests, multiplication entre signaux,...)

Ils sont la base des systèmes de compilation (génération automatique de code) :
- Simulink Real-Time Workshop (Matlab)
- Scicos (Inria)
- Lustre/Scade (Verimag/Esterel-Tecnologies)

Développement par modèles

Cette possibilité est à la base de la méthode de développement par modèles (Model-Based Development)

– Conception et validation sur modèles (Simulink, Scicos)
– Déploiement automatique par compilation

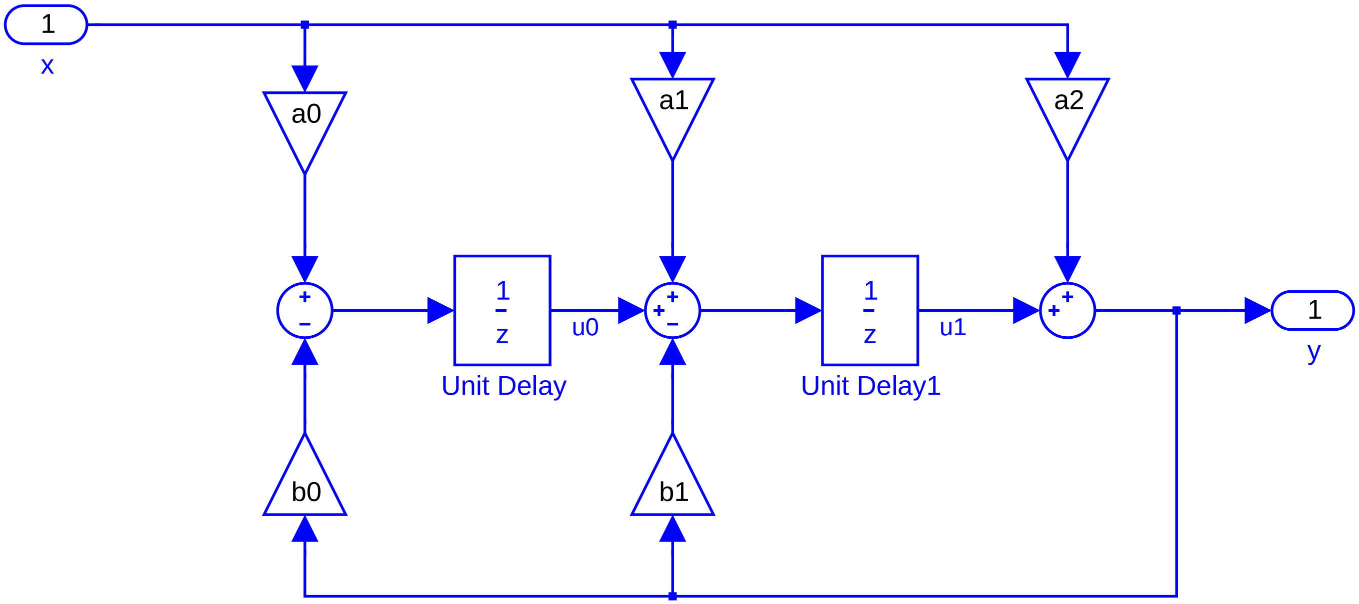
Cette méthode évite la phase de codage manuel, source d'erreur.

Elle garantit que ce qu'on a validé est ce qu'on exécute.

La mise en œuvre de cette méthode place le développement des systèmes d'automatismes et de traitement de signal à la pointe de l'informatique :

Dans ce domaine, l'informatique est véritablement passée de l'artisanat à l'industrie

Compilation



$$
\begin{aligned}
U_0(z) &= z^{-1}(a_0 X(z) - b_0 Y(z)) \\
U_1(z) &= z^{-1}(a_1 X(z) - b_1 Y(z) + U_0(z)) \\
Y(z) &= a_2 X(z) + U_1(z)
\end{aligned}
$$

Compilation générique

$$\begin{aligned}
U_0(z) &= z^{-1}(a_0 X(z) - b_0 Y(z)) \\
U_1(z) &= z^{-1}(a_1 X(z) - b_1 Y(z) + U_0(z)) \\
Y(z) &= a_2 X(z) + U_1(z)
\end{aligned}$$

Associer deux variables pour chaque variable d'état

– la valeur avant

– la valeur après

Compilation générique

$$
\begin{aligned}
U_0(z) &= z^{-1}(a_0 X(z) - b_0 Y(z)) \\
U_1(z) &= z^{-1}(a_1 X(z) - b_1 Y(z) + U_0(z)) \\
Y(z) &= a_2 X(z) + U_1(z)
\end{aligned}
$$

```
class deuxième_ordre {
      private :
        float a0, a1, a2, b0, b1;
        float u0, u1 ;
      public :
        deuxième_ordre(float a0, a1, a2, b0, b1;
               float u0, u1) {....}
        float compute(float x) {
          ...
          return y;
        }}
```

Compilation générique

$$
\begin{aligned}
U_0(z) &= z^{-1}(a_0 X(z) - b_0 Y(z)) \\
U_1(z) &= z^{-1}(a_1 X(z) - b_1 Y(z) + U_0(z)) \\
Y(z) &= a_2 X(z) + U_1(z)
\end{aligned}
$$

```
float compute(float x) {
  float y, up0, up1;
  y   = a2*x + u1 ;
  up0 = a0*x - b0*y ;
  up1 = a1*x - b1*y + u0;
  u0  = up0;
  u1  = up1;
  return y;
  }
}
```

Optimisations

Nombreuses possibilités grâce à la sémantique équationnelle des schémas-bloc.

- éliminer les buffers non utilisés
- supprimer des buffers en réordonnant les équations
- transformer le source grâce à la commutation des z^{-1}

Suppression de buffers

```
float compute(float x) {
  float y, up0, up1;
  y   = a2*x + u1 ;
  up0 = a0*x - b0*y ;
  up1 = a1*x - b1*y + u0;
  u0  = up0;
  u1  = up1;
  return y;
  }
}
```

```
float compute(float x) {
  float y, up0;
  y   = a2*x + u1 ;
  up0 = a0*x - b0*y ;
  u1  = a1*x - b1*y + u0;
  u0  = up0;
  return y;
  }
}
```

Réordonner les instructions

```c
float compute(float x) {
  float y, up0;
  y   = a2*x + u1 ;
  up0 = a0*x - b0*y ;
  u1  = a1*x - b1*y + u0;
  u0  = up0 ;
  return y;
  }
}
```

```c
float compute(float x) {
  float y;
  y   = a2*x + u1 ;
  u1  = a1*x - b1*y + u0;
  u0  = a0*x - b0*y ;
  return y;
  }
}
```

Commutation des z^{-1}

Grâce à la propriété :

Si f est une fonction statique (ne comportant pas d'opérateur z),

$$zf(x, y) = f(zx, zy)$$

C'est cette propriété qui nous a permis de réduire le nombre de retards dans le deuxième ordre.