

Petit cours d'automatique

- Pourquoi ce cours ?
- Modèle d'un système physique
- Résolution par transformée de Laplace
- Réponse en fréquence, spectre
- Commande par rétro-action
- Commande PID
- Simulation Matlab/Simulink
- Commande par ordinateur
- Transformée en Z
- Simulation Matlab/Simulink

Pourquoi ce cours ?

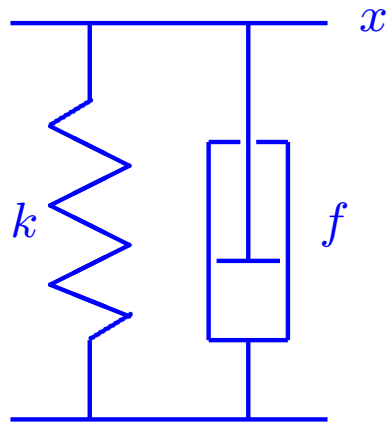
Les systèmes embarqués s'adressent à de nombreux domaines d'application et beaucoup de ces domaines traitent et contrôlent des données issues du monde physique :

1. il est important de savoir comment les spécialistes de ces domaines procèdent pour pouvoir coopérer avec eux ;
2. les ordinateurs interagissant avec les systèmes physiques forment des systèmes complexes qui acquièrent, par cette interaction, de nouvelles propriétés : des changements mineurs du point de vue informatique peuvent avoir des conséquences importantes du point de vue système ;
3. des langages et outils de simulation, d'origine automatique et traitement de signal, deviennent par extension, des outils de programmation dont l'importance croît ; certains sont devenus des standards de fait (avionique, automobile, ...). Il est important de connaître et comprendre cette

« informatique venue d'ailleurs »

Modèle d'un système physique

Une suspension de voiture



Forces en présence :

- ressort : $-k(x - x_0)$
- amortisseur : $-fx'$
- inertie : mx''
- force externe, pesanteur,...

bilan : équation différentielle

$$mx'' + fx' + kx = u$$

normalisation :

$$x'' + 2zwx' + w^2x = w^2u$$

- w pulsation propre
- z amortissement

Résolution par transformée de Laplace

$$X(s) = \mathcal{L}(x) = \int_0^{\infty} x(t)e^{-st} dt$$

propriétés :

– linéarité :

$$\mathcal{L}(ax + by) = a\mathcal{L}(x) + b\mathcal{L}(y)$$

– transforme les équations différentielles en équations algébriques :

$$\mathcal{L}(x') = s\mathcal{L}(x) - x(0)$$

– transforme les exponentielles en fractions rationnelles :

$$\mathcal{L}\left(\frac{t^n}{n!}e^{-\lambda t}\right) = \frac{1}{(s + \lambda)^{n+1}}$$

Application

$$x'' + 2zwx' + w^2x = w^2u \Rightarrow s^2X + 2zwsX + w^2X = w^2U$$

$$X = \frac{w^2}{s^2 + 2zws + w^2}U$$

Recherche de solutions :

- trouver les racines du dénominateur (pôles)
- décomposer en éléments simples et inverser la transformée.

Trois cas (réponse impulsionnelle $U = 1$) :

- $z^2 < 1$: racines complexes conjuguées :
- $z^2 = 1$: racine double
- $z^2 > 1$: racines réelles

- $z^2 < 1$: racines complexes conjuguées :

$$X = \frac{A}{s + zw + iw\sqrt{1 - z^2}} + \frac{B}{s + zw - iw\sqrt{1 - z^2}}$$

$$A = \frac{w^2}{-2iw\sqrt{1 - z^2}} \quad , \quad B = \frac{w^2}{2iw\sqrt{1 - z^2}}$$

$$x = \frac{w}{\sqrt{1 - z^2}} e^{-zwt} \frac{e^{iw\sqrt{1 - z^2}t} - e^{-iw\sqrt{1 - z^2}t}}{2i}$$

$$x = \frac{w}{\sqrt{1 - z^2}} e^{-zwt} \sin(w\sqrt{1 - z^2}t)$$

régime oscillant (suspension trop molle) !

- $z^2 = 1$: racine double

$$x = w^2 t e^{-wt}$$

- $z^2 > 1$: racines réelles

$$x = \frac{w}{\sqrt{z^2 - 1}} (e^{-w(z + \sqrt{z^2 - 1})t} - e^{-w(z - \sqrt{z^2 - 1})t})$$

(suspension trop dure) !

Réponse en fréquence, spectre

A la mémoire de Joseph Fourier

$$u = e^{-i\omega t}$$

$$U = \frac{1}{s + i\omega}$$

$$X = H \frac{1}{s + i\omega}$$

$$X = AH + B \frac{1}{s + i\omega}$$

$$B = H(-i\omega)$$

Les exponentielles complexes sont les vecteurs propres des opérateurs ;
 $H(-i\omega)$ est la valeur propre associée au vecteur propre $e^{-i\omega t}$.

Exemple de la suspension

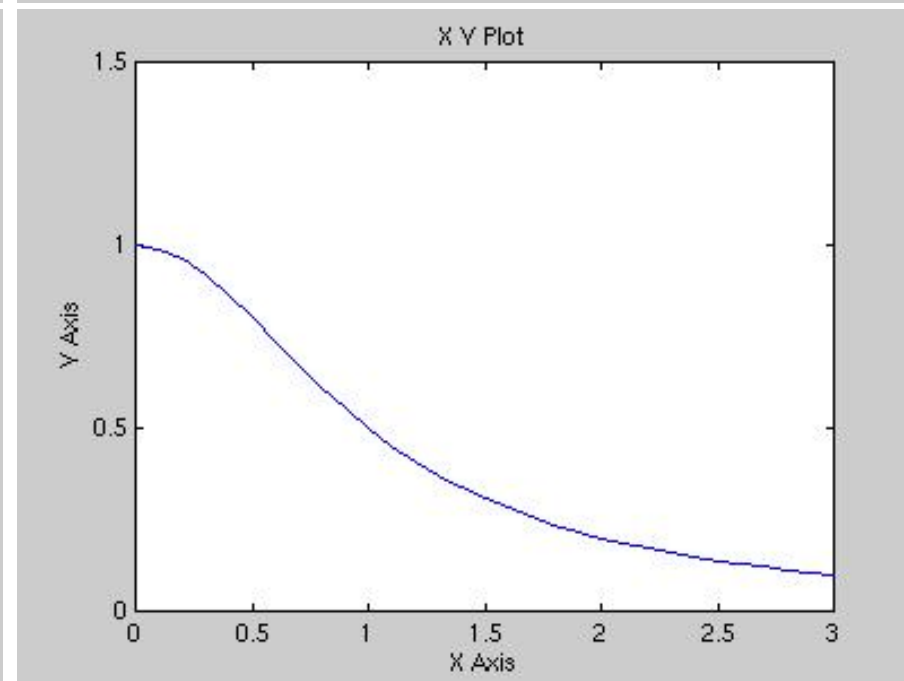
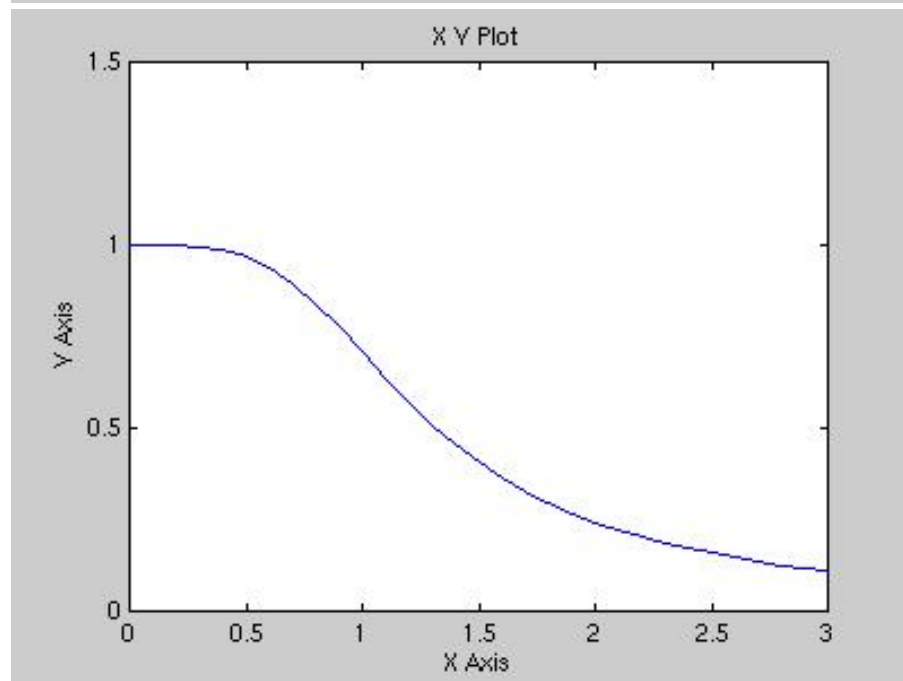
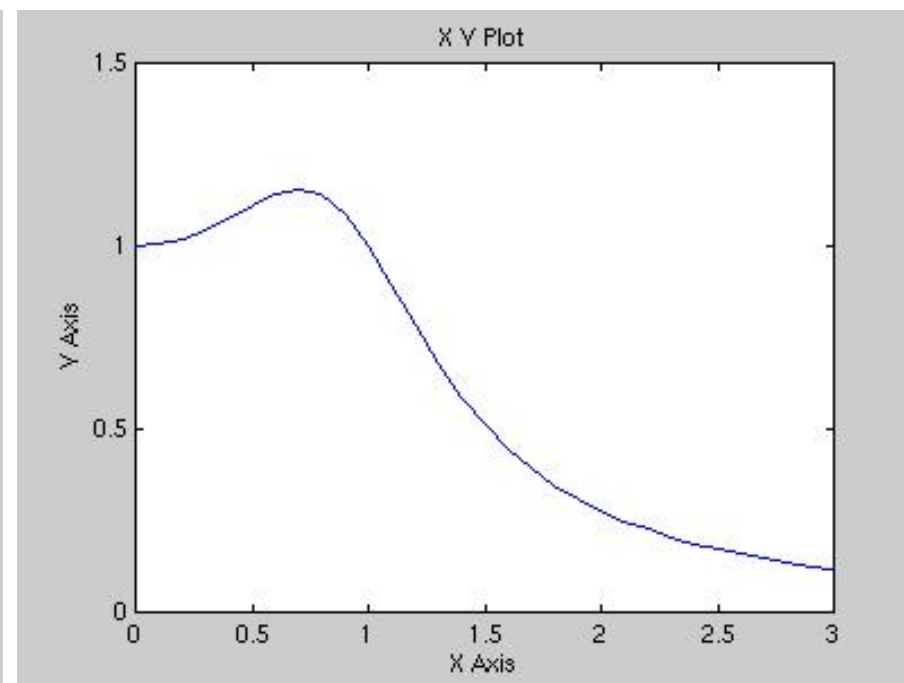
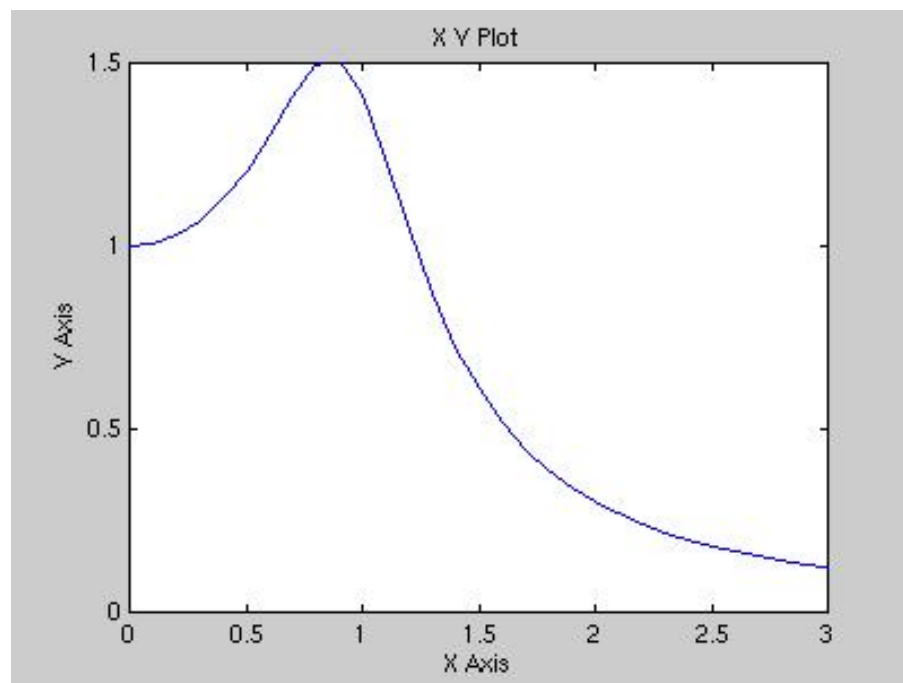
$$H = \frac{w_0^2}{s^2 + 2zw_0s + w_0^2}$$

$$H(-iw) = \frac{w_0^2}{-w^2 - 2izw_0w + w_0^2}$$

$H(-iw)$ est une fonction complexe de la fréquence d'excitation w Elle se décompose en amplitude et phase :

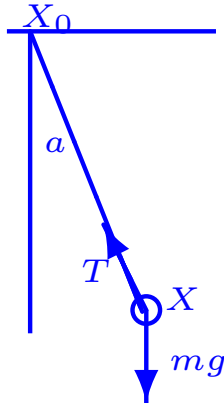
$$|H(-iw)| = \frac{w_0^2}{\sqrt{(w_0^2 - w^2)^2 + 4z^2w_0^2w^2}}$$
$$\text{Arg}H(-iw) = \text{Arctg} \frac{2zw_0w}{w_0^2 - w^2}$$

les pages suivantes décrivent le spectre d'amplitude en fonction de w pour $w_0 = 1$ et des valeurs croissantes de $z^2 = 0.5, 1, 2, 4$.



Commande par rétro-action

Le système à commander : grue mono-dimensionnelle, linéarisée



- pesanteur : $m \vec{g}$
- tension du cable : $k(\vec{X}_0 - \vec{X})$
- inertie : $m \vec{X}''$

$$m \vec{X}'' = k(\vec{X}_0 - \vec{X}) + m \vec{g}$$

Projection sur l'axe vertical :

$$m y'' = k(y_0 - y) - mg$$

$$0 = kl - mg$$

$$k = m \frac{g}{l}$$

Projection sur l'axe horizontal :

$$m x'' = k(x_0 - x)$$

$$x'' = \frac{g}{l} (x_0 - x)$$

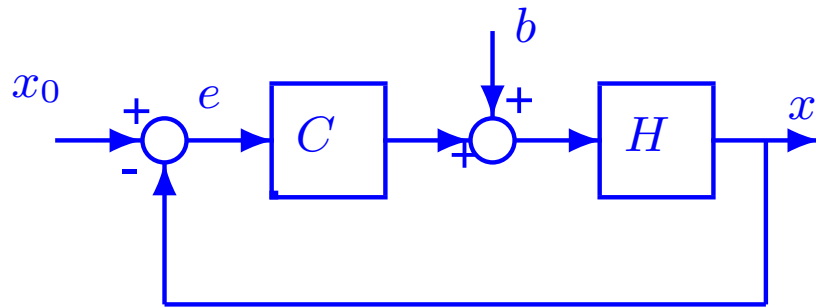
$$X = \frac{w^2}{s^2 + w^2} X_0$$

avec $w = \sqrt{\frac{g}{l}}$ pulsation propre du pendule.

aucun amortissement !

Notion de feedback

Commander en fonction de l'écart entre position désirée x_0 et obtenue x :



$$e = x_0 - x$$

$$x = H(Ce + b)$$

$$x = H(C(x_0 - x) + b)$$

$$(1 + HC)x = H(Cx_0 + b)$$

Calcul formel de ce diagramme !

$$x = \frac{HC}{1 + HC}x_0 + \frac{H}{1 + HC}b$$

$$x = \frac{HC}{1 + HC}x_0 + \frac{H}{1 + HC}b$$

Idée de base : C grand

$$\frac{HC}{1 + HC} \approx 1, \quad \frac{H}{1 + HC} \approx 0, \quad x \approx x_0$$

Mais attention au **pompage** :

$$\left. \begin{array}{l} H = \frac{w^2}{s^2 + 2zws + w^2} \\ C = k \end{array} \right\} \Rightarrow \frac{HC}{1 + HC} = \frac{w^2 k}{s^2 + 2zws + w^2(1 + k)}$$

$$w' = w\sqrt{1 + k} \quad z' = \frac{z}{\sqrt{1 + k}}$$

k grand : le système devient de plus en plus oscillant.

Commande PID

Pour réduire les oscillations, ajouter de la dérivée $k' s$

Pour rallier le but, ajouter de l'intégrale $\frac{k''}{s}$

$$C = k + k' s + \frac{k''}{s}$$

Application à la grue ($w = 1$)

$$\frac{HC}{1 + HC} = \frac{k' s^2 + k s + k''}{s^3 + k' s^2 + (k + 1)s + k''}$$

Placement des pôles

Choix des paramètres : identification à un dénominateur

$$s^3 + k's^2 + (k + 1)s + k'' = (s + 1)(s^2 + 1.4ws + w^2)$$

$$s^3 + k's^2 + (k + 1)s + k'' = s^3 + (1 + 1.4w)s^2 + (w^2 + 1.4w)s + w^2$$

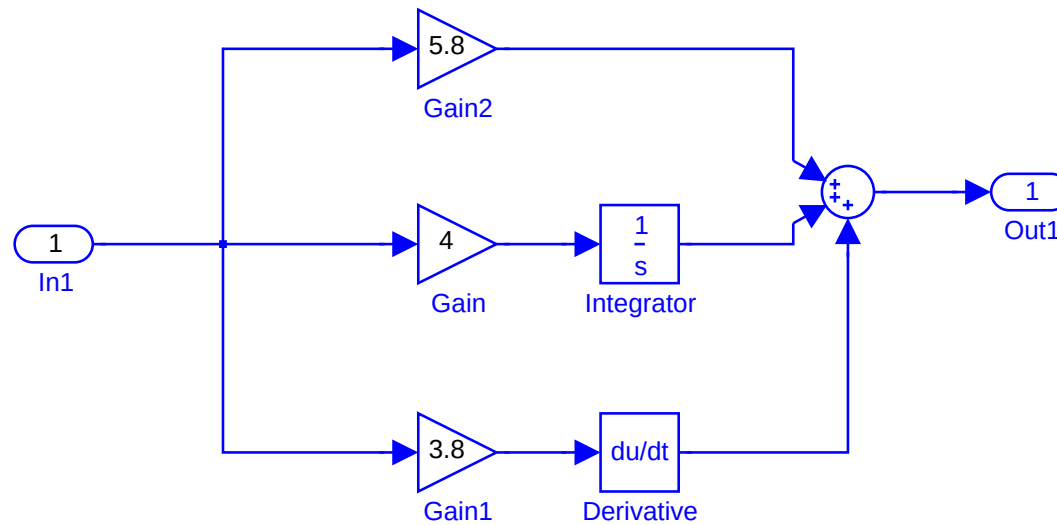
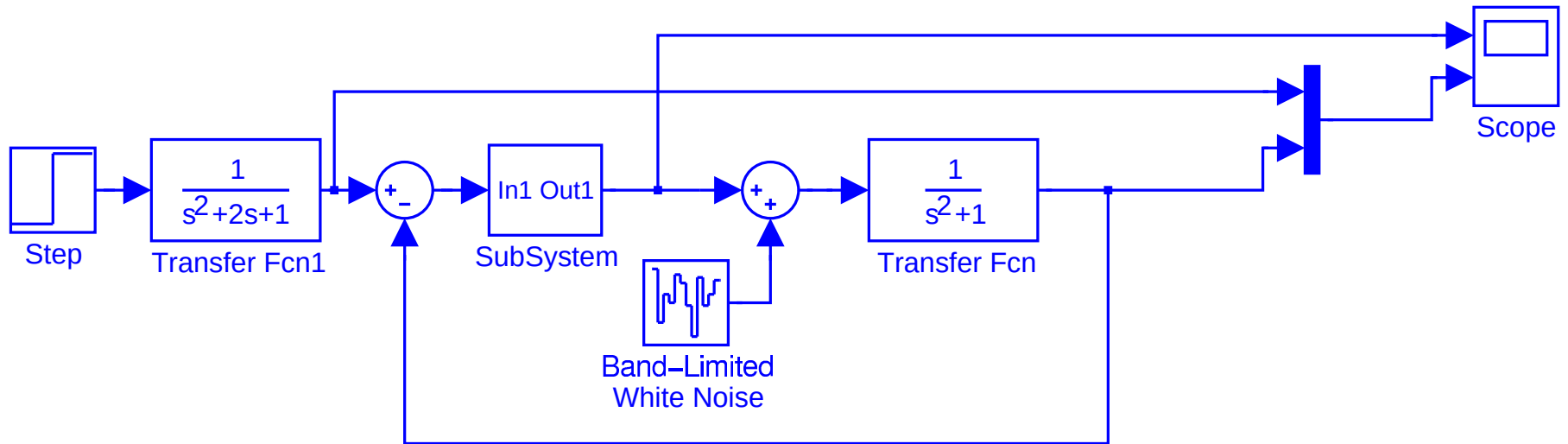
D'où

$$k'' = w^2, \quad k = w^2 + 1.4w - 1, \quad k' = 1 + 1.4w$$

Par exemple

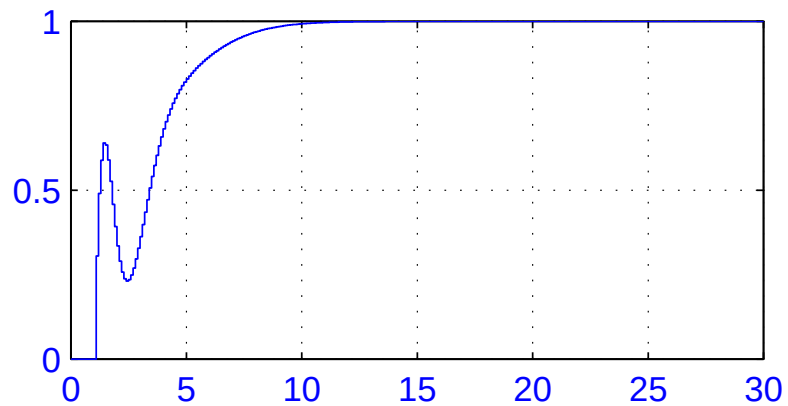
$$k'' = 4, \quad k = 5.8, \quad k' = 3.8$$

Simulation Matlab/Simulink

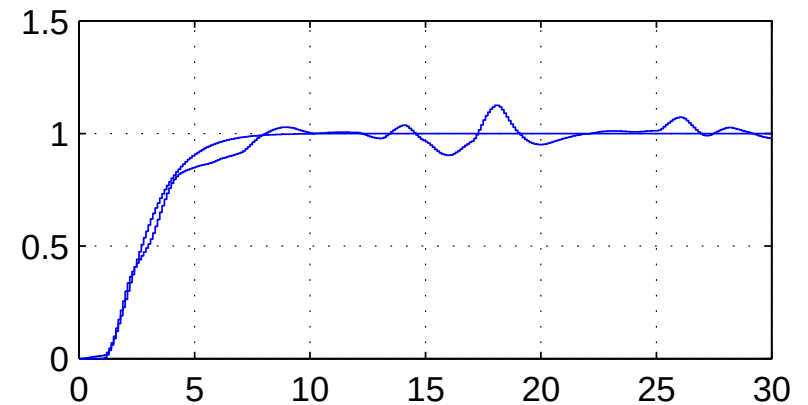
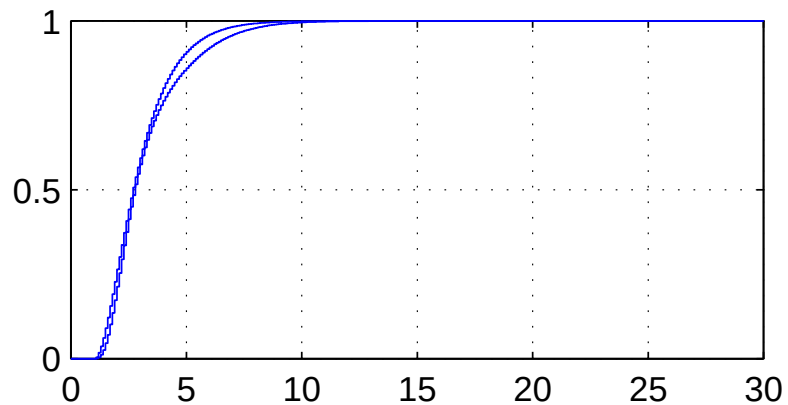
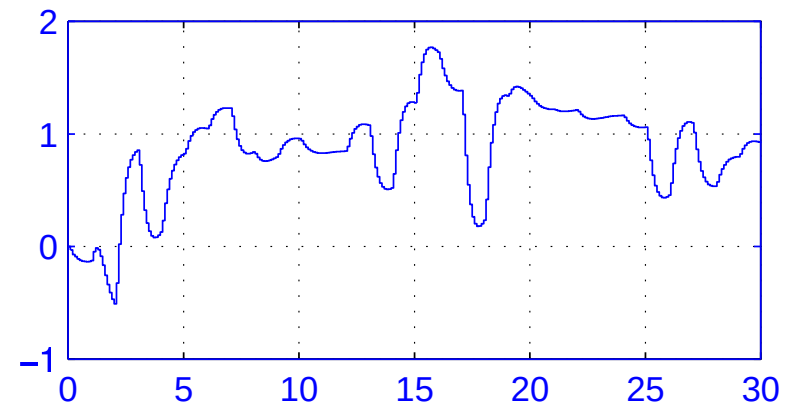


Résultats

Sans bruit



Avec bruit



Time offset: 0

Time offset: 0

Commande par ordinateur

Et les ordinateurs dans tout ça ?

idée générale : schémas d'intégration à pas constant

- échantillonnage et fonctionnement périodique

deux techniques :

- construire un contrôleur continu et l'échantillonner ;
- échantillonner le système à contrôler et construire un contrôleur échantillonné (théorie de la commande échantillonnée).

Dans tous les cas, il faut choisir la période.

Commande échantillonnée ---

Fondée sur le parallèle

	Continu	Echantillonné
Modèle	équations différentielles	équations aux différences
Solutions	exponentielles	puissances
Transformée	$\mathcal{L}(f) = \int_0^\infty f(t)e^{-st}dt$	$\mathcal{Z}(f) = \sum_0^\infty f(n)z^{-n}$
Intégrale	s^{-1}	$\frac{z}{z-1}$
Retard	e^{-Ts}	z^{-1}

Propriétés de la transformée en z _____

- linéarité :

$$\mathcal{Z}(ax + by) = a\mathcal{Z}(x) + b\mathcal{Z}(y)$$

- transforme les équations aux différences en équations algébriques :

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

- transforme les puissances en fractions rationnelles :

$$\mathcal{Z}(C_n^k a^{n-k}) = \left(\frac{z}{z-a} \right)^{k+1}$$

Choix de la période

Solution théorique :

Théorème de Shannon : la fréquence d'échantillonnage doit être supérieure à deux fois la fréquence de coupure.

Solution pratique : analyse numérique

Théorème fondamental de l'analyse numérique

Théorème des accroissements finis : Pour tout x, t , il existe $\alpha \in [0, 1]$ tel que

$$x(t + T) = \sum_0^k \frac{T^i}{i!} x^{(i)}(t) + \frac{T^{k+1}}{(k+1)!} x^{(k+1)}(t + \alpha T)$$

Exemple : schéma d'Euler

$$x(t + T) \approx x(t) + Tx'(t)$$

car

$$|x(t + T) - (x(t) + Tx'(t))| \leq \frac{T^2}{2} \sup_t |x''(t)|$$

Application : exponentielle

$$x(t) = e^{-at} \quad x''(t) = a^2 e^{-at} \leq a^2$$

$$\frac{T^2 a^2}{2} \leq 0.01 \Rightarrow T \leq \frac{0.14}{a}$$

En pratique on prend

$$T = \frac{0.1}{a}$$

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

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

Et Euler ?

Appliquons à une fonction de transfert H :

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

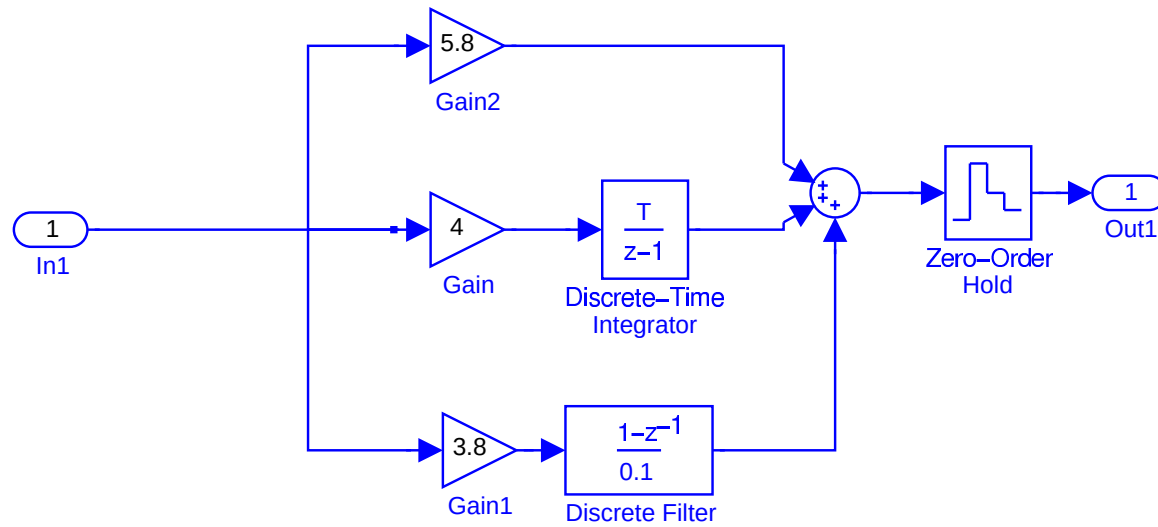
On est passé d'un système continu à un système échantillonné, implantable sur ordinateur

Cela correspond à la méthode d'Euler :

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

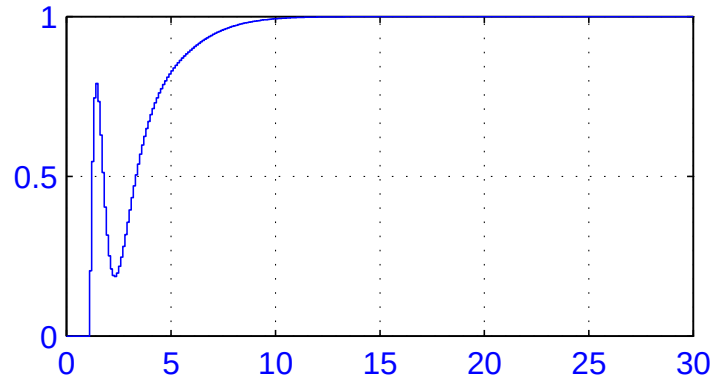
$$x'(t) \approx \frac{x(n) - x(n - 1)}{T}$$

Simulation Matlab/Simulink

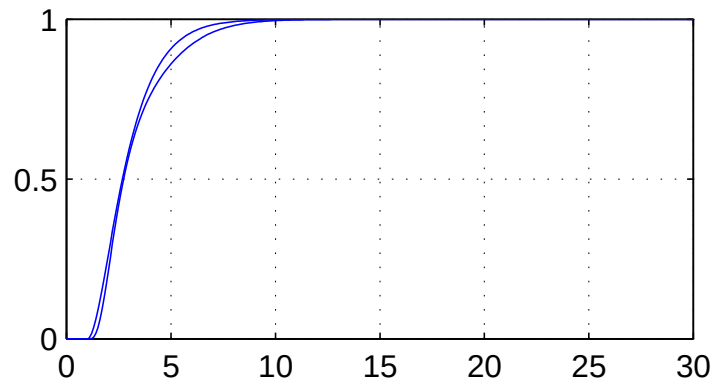
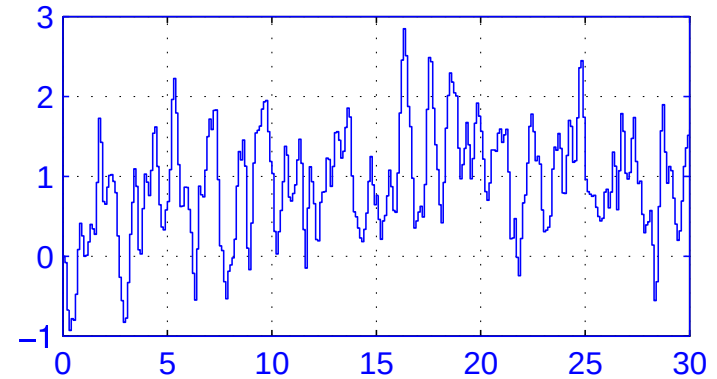


Résultats

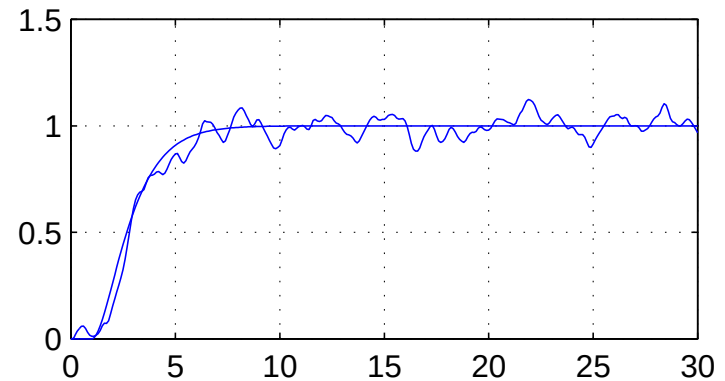
Sans bruit



Avec bruit



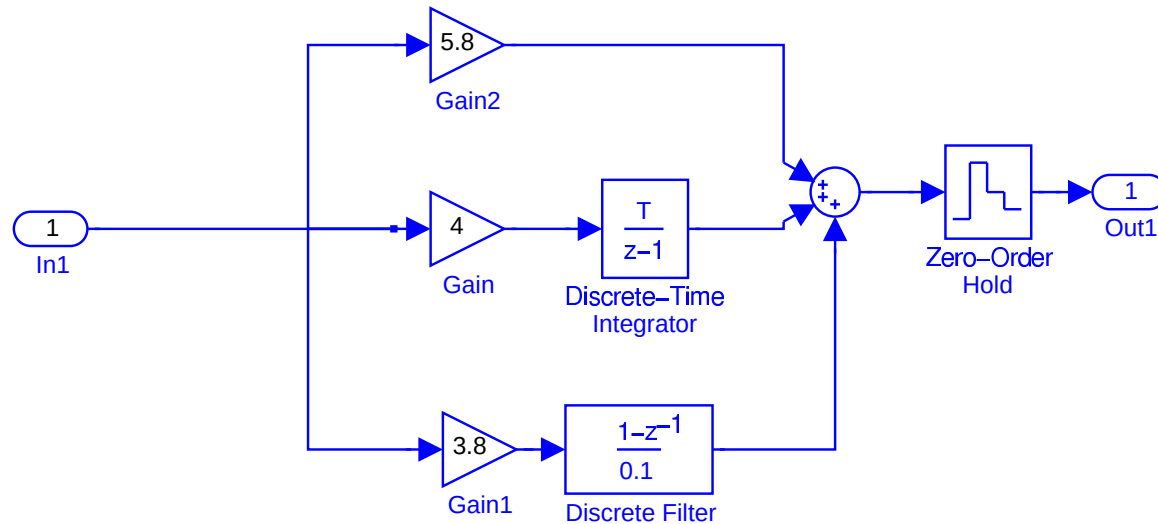
Time offset: 0



Time offset: 0

comparables à ceux obtenus en continu

Z et Lustre



Le même en Lustre :

```
y  = 5.8*x + 4.0*yi + 3.8*yd ;  
yi = T*x + (0.0 -> pre yi) ;  
yd = (x - (0.0 -> pre x))/T ;
```

De Z à Lustre

En transformée en Z

$$Y_i = \frac{Tz}{z-1}X$$

D'où

$$(z-1)Y_i = TzX$$

$$zY_i = TzX - Y_i$$

$$Y_i = TX - \frac{1}{z}Y_i$$

Le même en Lustre :

```
yἷ = T*x + (0.0 -> pre yἷ) ;
```

Conclusion

On note la parenté (filiation) entre Lustre et les méthodes de l'automatique :

Tous deux réfléchissent en termes de réseaux d'opérateurs agissant globalement sur des signaux.

L'automatique linéaire a un calcul symbolique plus riche (fractions rationnelles) mais limité au linéaire.

Lustre a aussi un calcul symbolique bien que plus restreint (commutation des pre)

Cela explique sans doute la bonne acceptation de Lustre dans le monde de l'automatique.