



UNIVERSITÉ  
DE LORRAINE



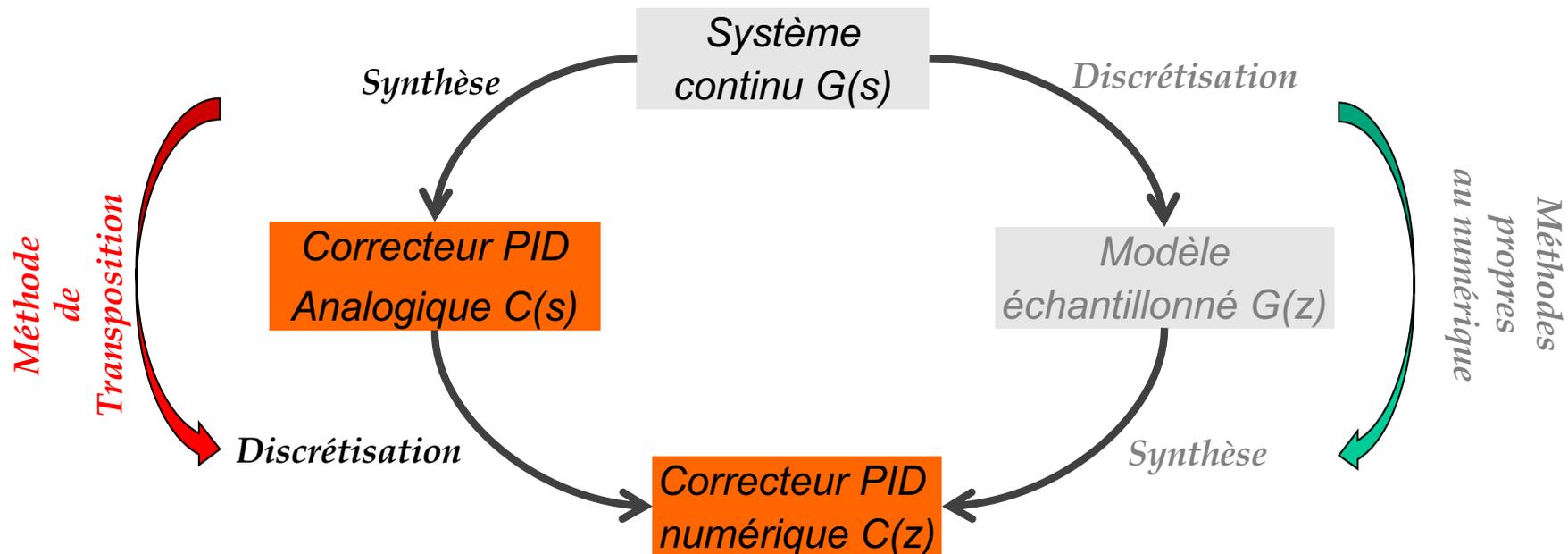
POLYTECH<sup>®</sup>  
NANCY

*Synthèse d'un  
correcteur PID numérique  
par transposition du PID analogique*

Hugues GARNIER

[hugues.garnier@univ-lorraine.fr](mailto:hugues.garnier@univ-lorraine.fr)

## Les 2 voies pour la synthèse de correcteurs numériques

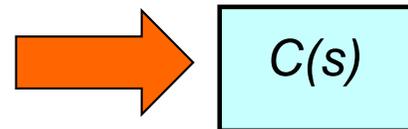


- La synthèse de correcteurs PID numériques par transposition de correcteurs PID analogiques est une approche couramment utilisée dans le domaine industriel pour deux raisons majeures :
  - les méthodes de synthèse de correcteurs PID analogique sont généralement bien maîtrisées
  - les spécifications sont plus facilement interprétables avec des modèles continus qu'avec des modèles échantillonnés

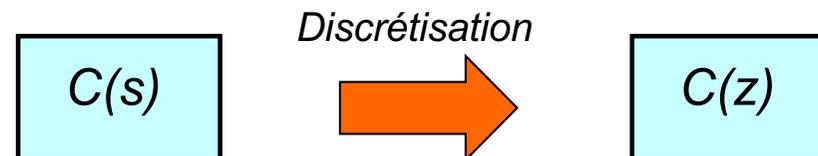
## Synthèse de correcteur numérique par transposition du correcteur analogique

- Méthodologie

1. Synthèse d'un correcteur PID continu  $C(s)$  par une des méthodes de synthèse traditionnelles (Ziegler-Nichols ou autres) déterminé à partir du modèle du système à contrôler permettant de respecter le cahier des charges



2. Transposition de la fonction de transfert du PID continu  $C(s)$  en un correcteur numérique  $C(z)$  pour avoir un algorithme de PID numérique qui s'approche le plus possible de comportement du contrôle numérique



## PID parallèle

Dans le domaine temporel

$$u(t) = k_p \varepsilon(t) + k_i \int_0^t \varepsilon(\tau) d\tau + k_d \frac{\varepsilon(t)}{dt}$$

Dans le domaine de Laplace

$$U(s) = \left( k_p + \frac{k_i}{s} + k_d s \right) \varepsilon(s)$$

$$C(s) = k_p + \frac{k_i}{s} + k_d s$$

$$K_p = k_p \quad T_i = \frac{k_p}{k_i} \quad T_d = \frac{k_d}{k_p}$$

## PID idéal rencontré en pratique

Dans le domaine temporel

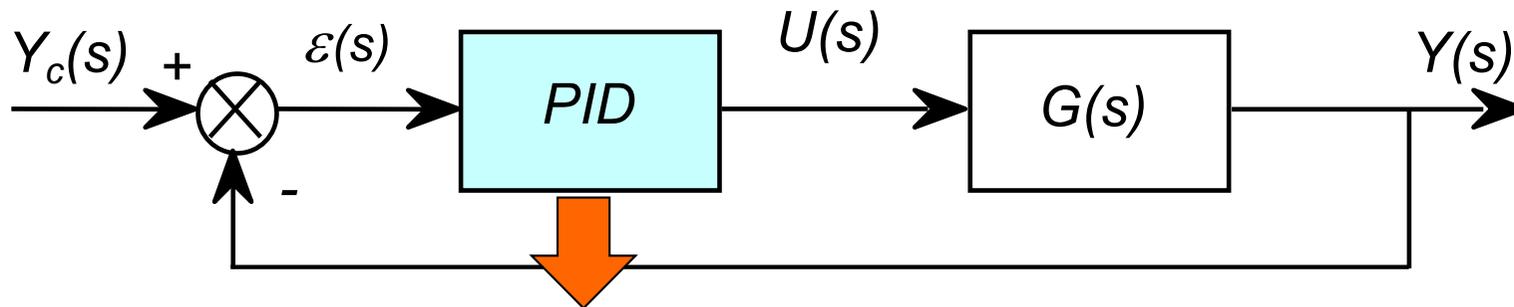
$$u(t) = K_c \left( \varepsilon(t) + \frac{1}{T_i} \int_0^t \varepsilon(\tau) d\tau + T_d \frac{\varepsilon(t)}{dt} \right)$$

Dans le domaine de Laplace

$$U(s) = K_p \left( 1 + \frac{1}{T_i s} + T_d s \right) \varepsilon(s)$$

$$C(s) = K_p \left( 1 + \frac{1}{T_i s} + T_d s \right)$$

## Rappel sur les correcteurs PID



$$u(t) = K_p \left( e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{de(t)}{dt} \right)$$

*PID idéal* Correcteur  $C(s) = K_p \left( 1 + \frac{1}{T_i s} + T_d s \right)$

*PID réel* Avec dérivée filtrée  $C(s) = K_p \left( 1 + \frac{1}{T_i s} + \frac{T_d s}{1 + T_d s / N} \right) \quad (N \geq 5)$

Réglage simple et adapté à la plupart des systèmes

- PI : 90%, PID : 95%
- Réglage souvent itératif (essai / erreur)

## Version numérique du PID par les méthodes de transposition usuelles

Approximation avancée  $C(z) = K_p \left[ 1 + \frac{1}{T_i} \frac{T_e z}{z-1} + \frac{N(z-1)}{\left(1 + \frac{NT_e}{T_d}\right) z - 1} \right]$

$$s = \frac{z-1}{T_e} = \frac{1-z^{-1}}{T_e z^{-1}}$$

Approximation retardée  $C(z) = K_p \left[ 1 + \frac{1}{T_i} \frac{T_e}{z-1} + \frac{N(z-1)}{z - \left(1 - \frac{NT_e}{T_d}\right)} \right]$

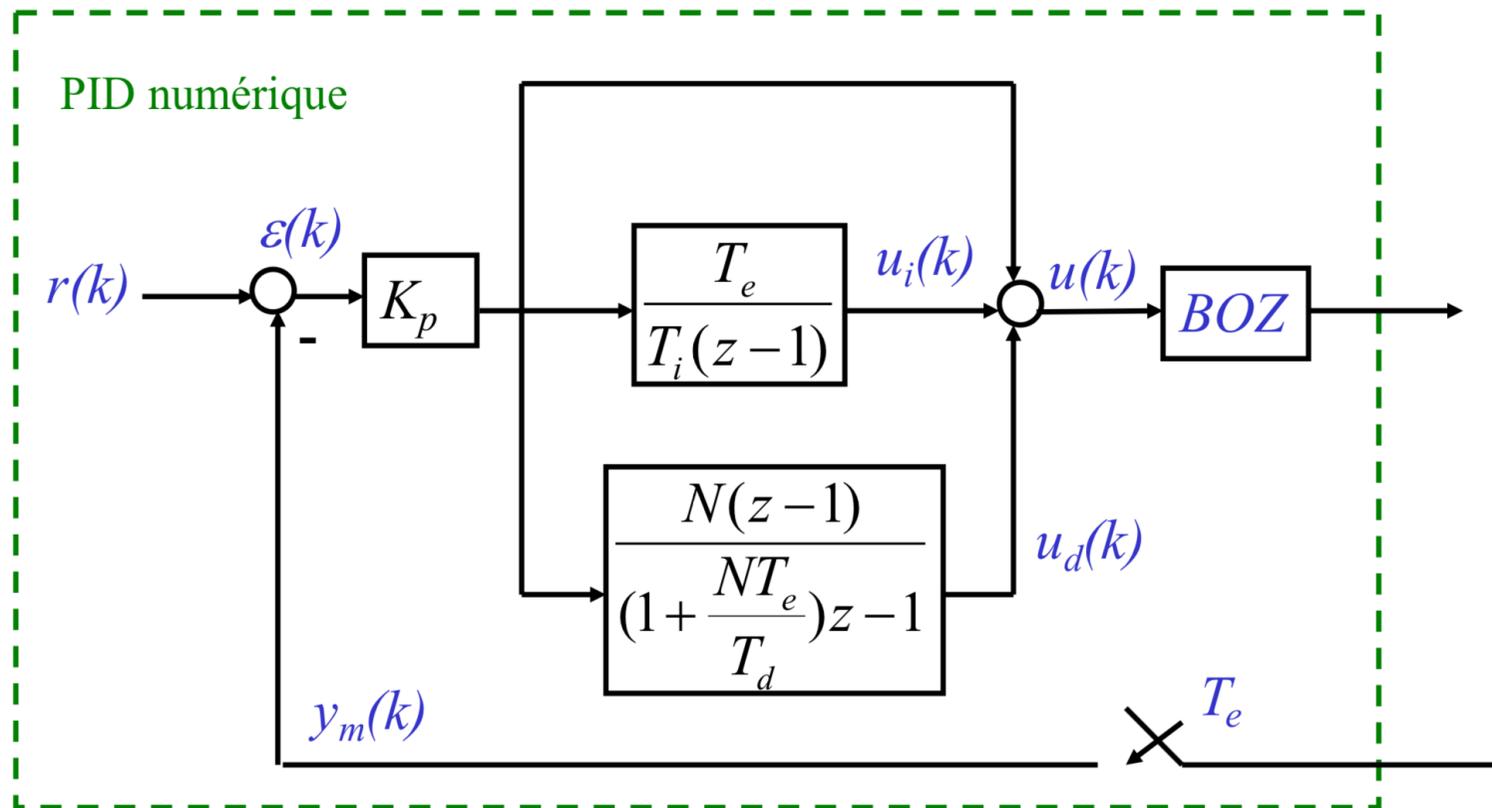
$$s = \frac{z-1}{T_e z} = \frac{1-z^{-1}}{T_e}$$

Approximation de Tustin  $C(z) = K_p \left[ 1 + \frac{T_e}{2T_i} \frac{z+1}{z-1} + \frac{N(z-1)}{\left(1 + \frac{NT_e}{2T_d}\right) z - \left(1 - \frac{NT_e}{2T_d}\right)} \right]$

$$s = \frac{2z-1}{T_e(z+1)} = \frac{2(1-z^{-1})}{T_e(1+z^{-1})}$$

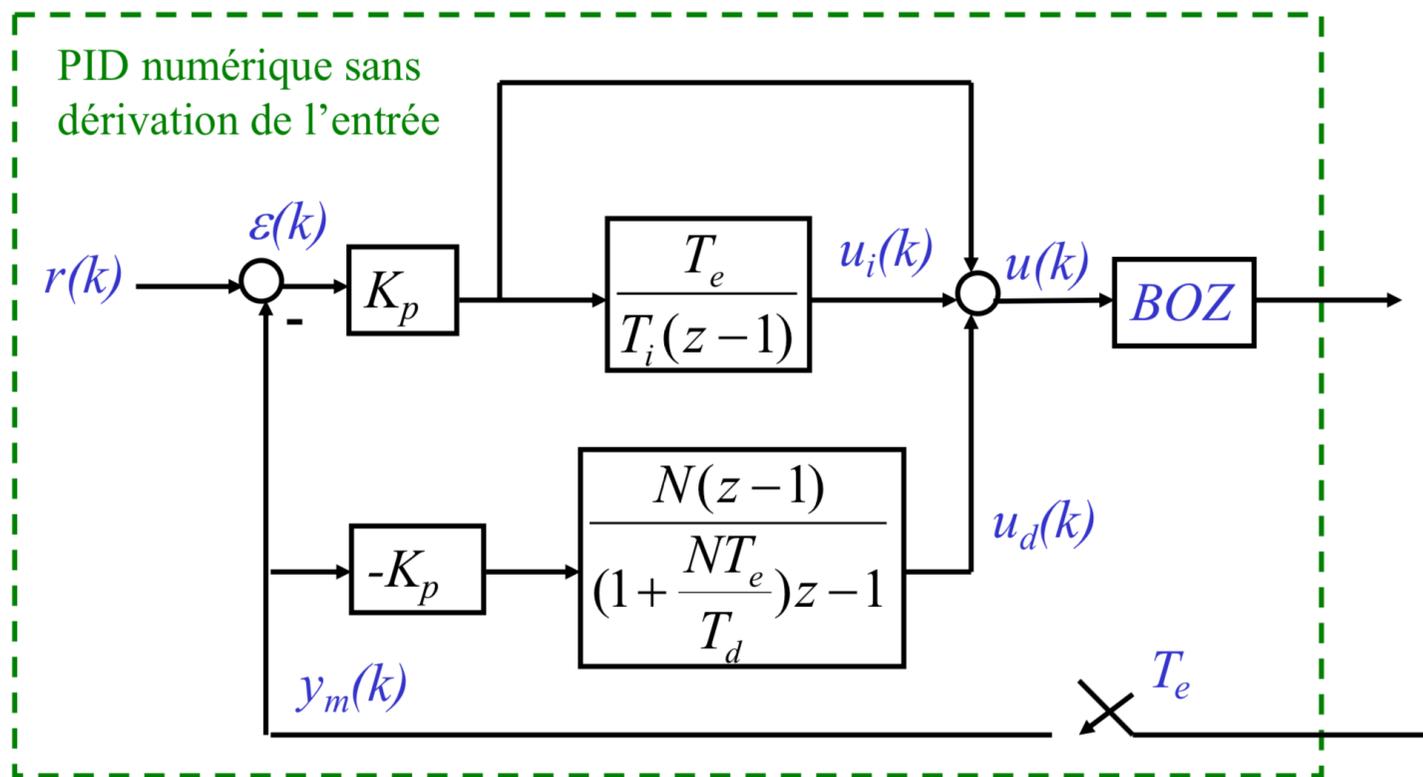
Approximation retardée : *formules plus simples souvent utilisées en pratique pour cette raison*

## Schéma d'implantation de la forme standard d'un PID numérique (*approximation retardée*)



## Schéma d'implantation de la forme sans dérivation de l'entrée d'un PID numérique (*approximation retardée*)

En pratique, on dérive rarement le terme de consigne pour éviter des variations brutales de la commande lors de changement brusque de type échelon sur la consigne. Le schéma devient alors



Algorithme d'implantation de la forme sans dérivation de l'entrée d'un PID numérique (*approximation retardée*)

$$\varepsilon(k) = r(k) - y_m(k)$$

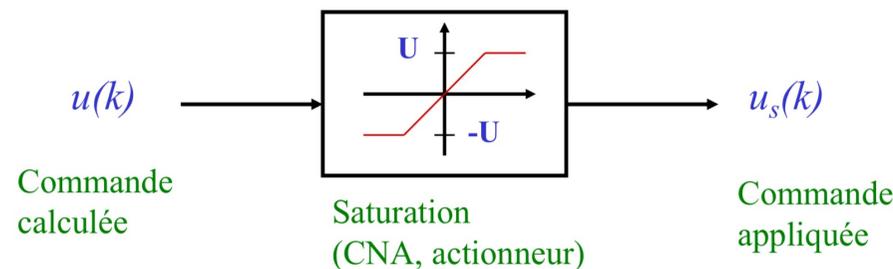
$$u(k) = K_p \varepsilon(k) + u_i(k) + u_d(k)$$

$$u_i(k) = u_i(k-1) + K_p \frac{T_e}{T_i} \varepsilon(k-1)$$

$$u_d(k) = \frac{1}{1 + \frac{NT_e}{T_d}} u_d(k-1) - \frac{K_p N}{1 + \frac{NT_e}{T_d}} [y_m(k) - y_m(k-1)]$$

## Anti-saturation d'intégrale

- La partie intégrale peut entraîner des effets indésirables lorsque, en raison d'un signal d'erreur trop grand, l'intégrateur sature
- L'actionneur reste alors en butée, même lorsque la sortie du système varie

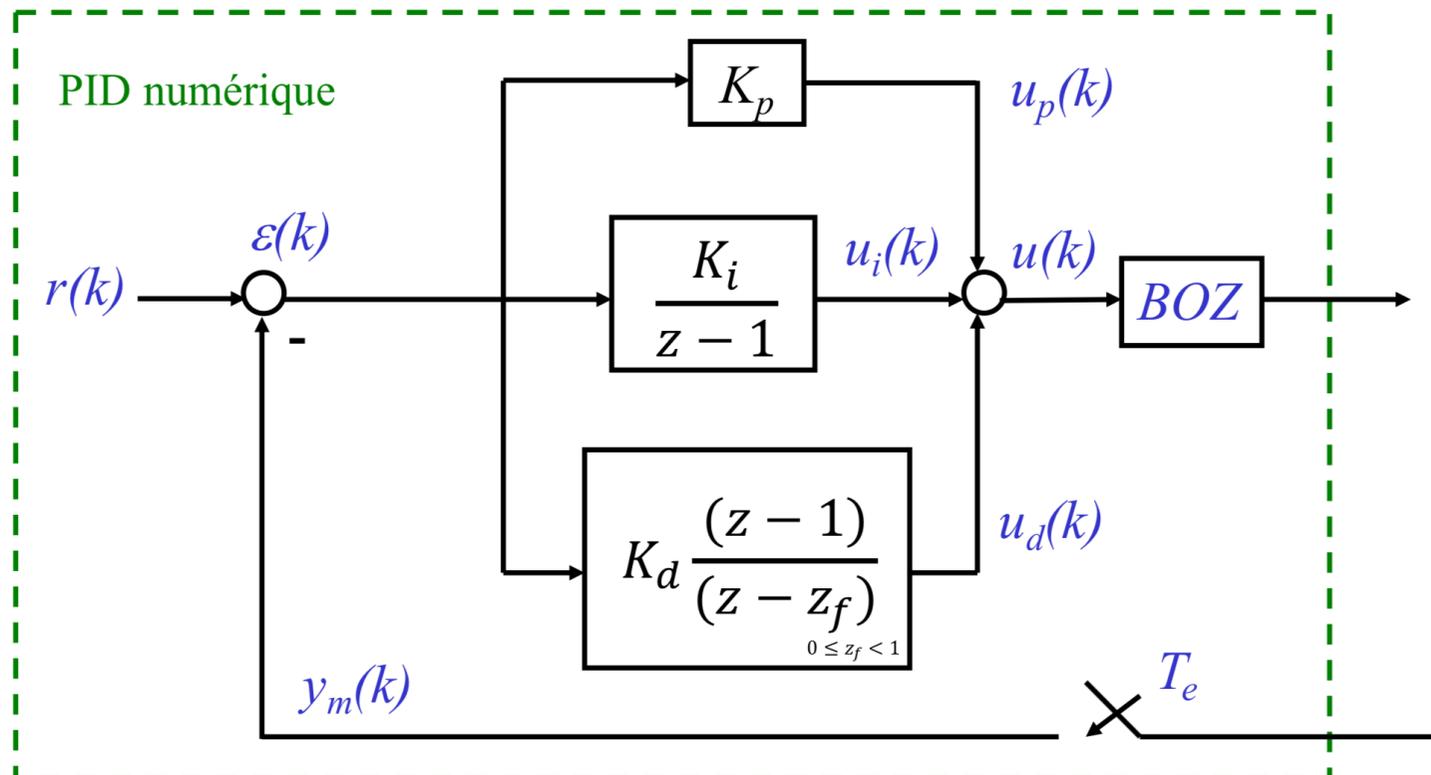


$$u_s(k) = \begin{cases} U & \text{si } u(k) > U \\ u(k) & \text{si } |u(k)| \leq U \\ -U & \text{si } u(k) < -U \end{cases}$$

- Une approche possible pour éliminer cet effet consiste à introduire un bouclage sur l'intégrateur, ramenant l'écart entre l'entrée  $u(k)$  et la sortie  $u_s(k)$  de la saturation (réelle ou simulée), avec une constante d'intégration  $T_t$

## Forme parallèle du PID numérique : un réglage plus pratique

$$C(s) = K_p \left( 1 + \frac{1}{T_i s} + T_d s \right) = K_p + \frac{K_i}{s} + K_d s$$



## PID numérique dans un code informatique

```
# def Te, Ki, Kp, Kd, r
i = 0
e0 = e = 0
while running:
    # update measure
    y = ...
    e = r - y
    # update integral
    i += Ki*Te*e
    # compute command
    u = Kp * (e + i + Kd/Te*(e-e0))
    # save current error
    e0 = e
```

## Réglage des actions du correcteur PID analogique

### Rappels

- Le réglage du correcteur PID analogique passe par le choix de

$$K_p, T_i, T_d, T_t, N$$

- $N$  est souvent fixé à  $N=10$
  - $T_t$  est choisie dans la plage  $[0,1T_i; T_i]$
- Pour la détermination des paramètres  $K_p, T_i, T_d$  des méthodes de réglage ont été proposées comme par exemple celles de Ziegler-Nichols
- Ces réglages constituent une base qu'il convient d'affiner en fonction des performances désirées

## Influence des actions P, I et D dans le cas de la forme parallèle du PID numérique

$$C(s) = K_p \left( 1 + \frac{1}{T_i s} + T_d s \right) = K_p + \frac{K_i}{s} + K_d s$$

Gain	T montée	T stabilisation	Dépassement	Erreur statique
$K_p \nearrow$	Diminue	Augmente	Augmente	Diminue
$K_i \nearrow$	Diminue	Augmente	Augmente	Annule
$K_d \nearrow$	–	Diminue	Diminue	–

Démarche : partir d'un premier jeu de gains

- En simulation (système approximé) ou sur système réel
- Préréglage sur réponse indicielle
- Préréglage sur système bouclé

Le préréglage est parfois présenté comme un auto-réglage

Mais souvent c'est vraiment un **préréglage**...

## Régulation numérique par transposition d'un correcteur continu – A retenir

- Méthodes de transposition conseillées en pratique :
  - Approximation bilinéaire (Tustin) ou approximation retardée (car formules les plus simples)
- Comportement de la régulation numérique au mieux équivalent à celui de la régulation analogique mais souvent moins bon
  - convient si le système est lent
  - convient si la période d'échantillonnage est rapide par rapport à la dynamique principale  $T_c$  du correcteur (si  $T_e < T_c/10$ )
- Attention ! Même si la stabilité du système bouclé avec le correcteur analogique est vérifiée, cela ne garantit pas la stabilité du système bouclé avec le correcteur numérique !
  - Il faut en particulier vérifier que la période d'échantillonnage choisie n'entraîne pas une perte de la stabilité du système bouclé
  - L'effet de la présence du bloqueur d'ordre zéro (retard additionnel) n'est pas pris en compte dans la synthèse et peut aussi jouer sur la stabilité

# Perspectives pour le cours d'Automatique

## *Réservées pour ceux qui vont choisir le parcours SIA*

- Stratégies de commande s'appuyant sur un modèle d'état
  - Commande par retour d'état : nouveau paradigme s'appuyant sur un modèle d'état (3A-S6)
  - Observateur et estimateur pour la commande (4A)
  - Commande optimale et filtrage de Kalman (4A et 5A)
- Stratégies de modélisation et de commande orientées IA s'appuyant sur l'exploitation de données
  - Apprentissage de modèles dynamiques (4A)
- Applications en robotique mobile (4A et 5A)
- Aspects non abordés (*par manque de temps*) :
  - Commande prédictive, commande adaptative, commande non linéaire, ...