

**MASTER D'UNIVERSITE SPECIALISE
INGENIERIE DES SYSTEMES EMBARQUES**

**Cours Capteurs et Régulation
Industrielle**

**CH_7: Les régulateurs PID
industriels**

Par:

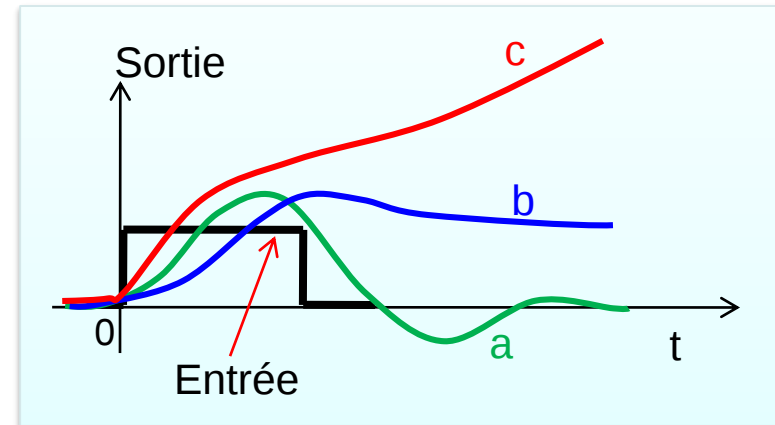
Abdellah LASSIOUI

1. Notions de stabilité, rapidité, dépassement et précision

1.1. Stabilité:

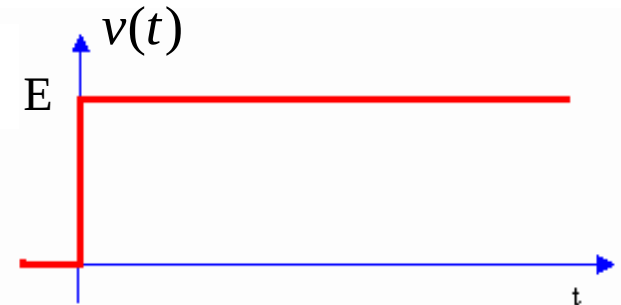
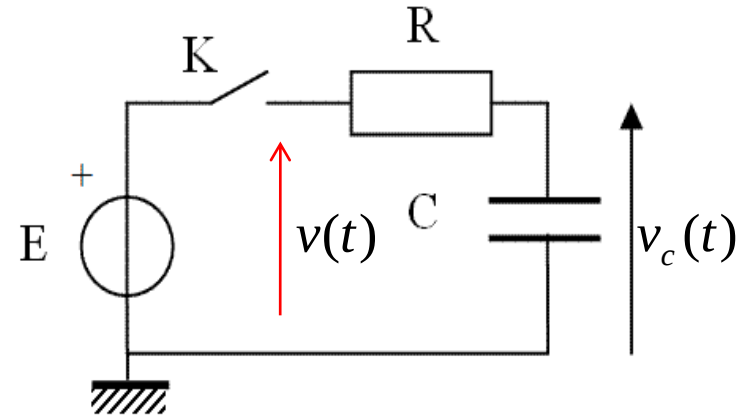
● **Définition** : Un procédé asservi ou non est stable si à une variation bornée du signal d'entrée correspond une variation bornée du signal de sortie. Une variation d'un signal est dite bornée lorsqu'elle est constante en régime permanent.

- *a: la sortie tend vers zéro: le système est asymptotiquement stable.*
- *b: la sortie tend vers une valeur finie: le système est à la limite de stabilité (il est marginalement stable).*
- *c: la sortie diverge: le système est instable.*



1.2. Rapidité:

- **Signal échelon:** un **échelon de tension** est un signal électrique nul avant un instant t_0 , et de tension constante après cet instant.
- **Réponse indicielle:** c'est la réponse correspondant à une entrée de type échelon.

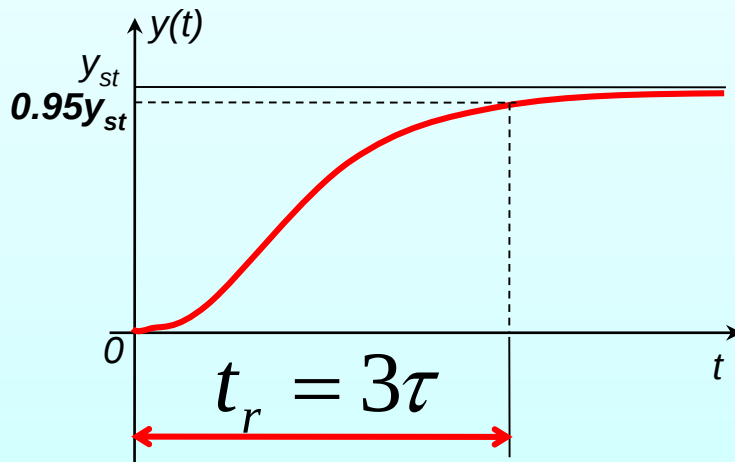


1.2. Rapidité:

$$\Omega \text{ (rad/s)} = K \text{ (SI)} * E \text{ (V)}$$

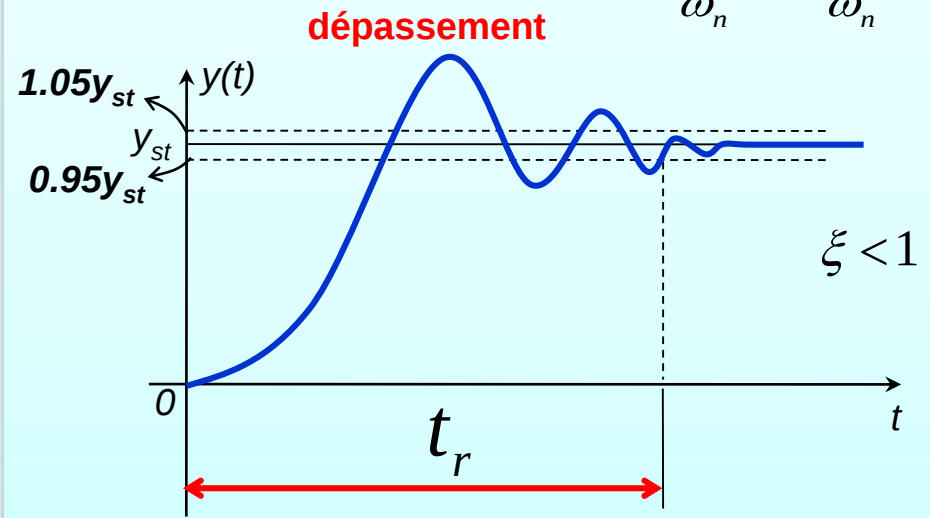
- **Temps de réponse à 5%:** C'est le temps nécessaire pour que la réponse entre dans la bande $\pm 5\%$ de la valeur finale.
- Il mesure la rapidité d'un système

Cas d'un 1^{er} ordre: $G(p) = \frac{y(p)}{u(p)} = \frac{K}{1 + \tau p}$



Réponse indicielle apériodique

Cas d'un 2^{ème} ordre: $G(p) = \frac{K}{1 + \frac{2\xi}{\omega_n} p + \frac{p^2}{\omega_n^2}}$

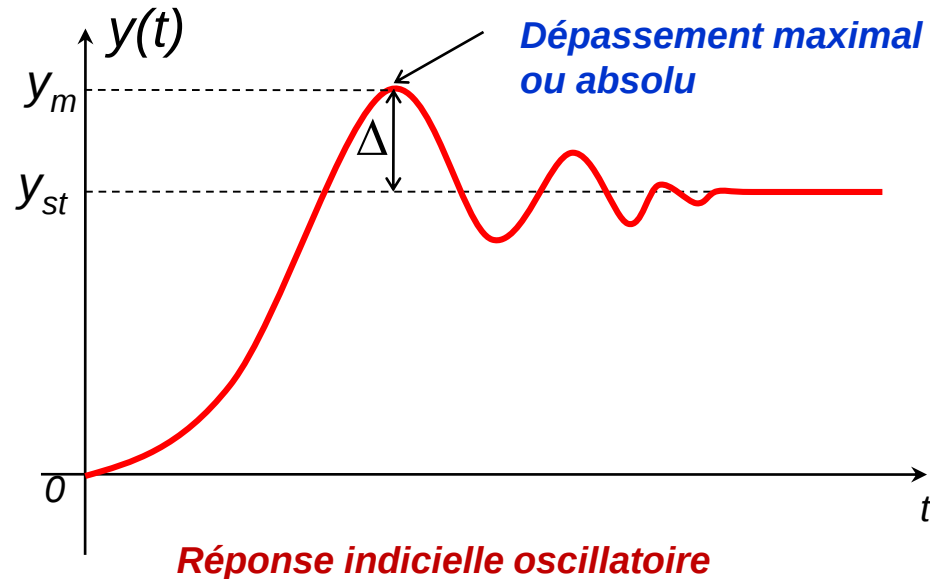


Réponse indicielle oscillatoire

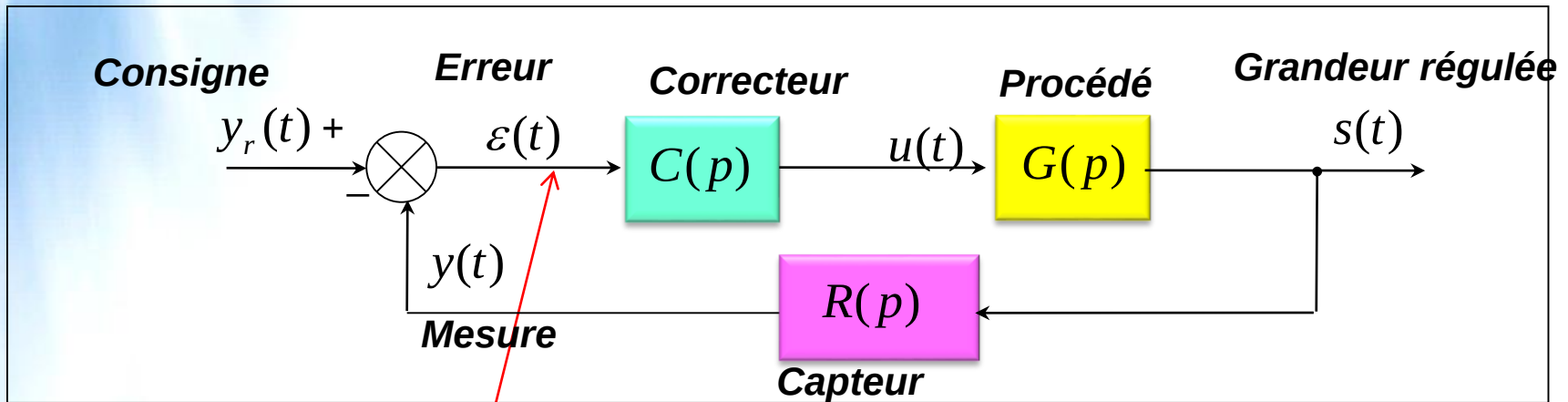
1.3. Dépassement relatif D

$$D = \frac{\Delta}{y_{st}} \times 100 \quad (\text{en } \%)$$

$$\Delta = y_m - y_{st}$$



1.4. Précision:



- La Précision d'un système asservi s'appréhende par l'étude de la grandeur d'erreur.
- Une précision statique : correspond à l'erreur permanente observée.
- En général, nous avons besoin que cette erreur soit nulle en régime permanent.

2. Objectifs d'un régulateur:

- Un régulateur est conçu pour que l'ensemble corrigé satisfasse un **cahier des charges**:
 - ▶ Meilleure stabilité,
 - ▶ Temps de réponse réduit,
 - ▶ Bonne précision statique,
 - ▶ Dépassement maximal contrôlé,
 - ▶

3. Les différentes actions d'un régulateur:

- Un régulateur contient le plus souvent une combinaison des actions suivantes :
- Une action **Proportionnelle (P)** : principalement utilisée pour améliorer la *rapidité*.
- Une action **Intégrale (I)** : principalement utilisée pour améliorer la *précision*.
- Une action **Dérivée (D)** : principalement utilisée pour améliorer la *stabilité*.

3.1. Action proportionnelle (P):

Equation:

$$u(t) = K_p \cdot \varepsilon(t)$$

Fonction de transfert:

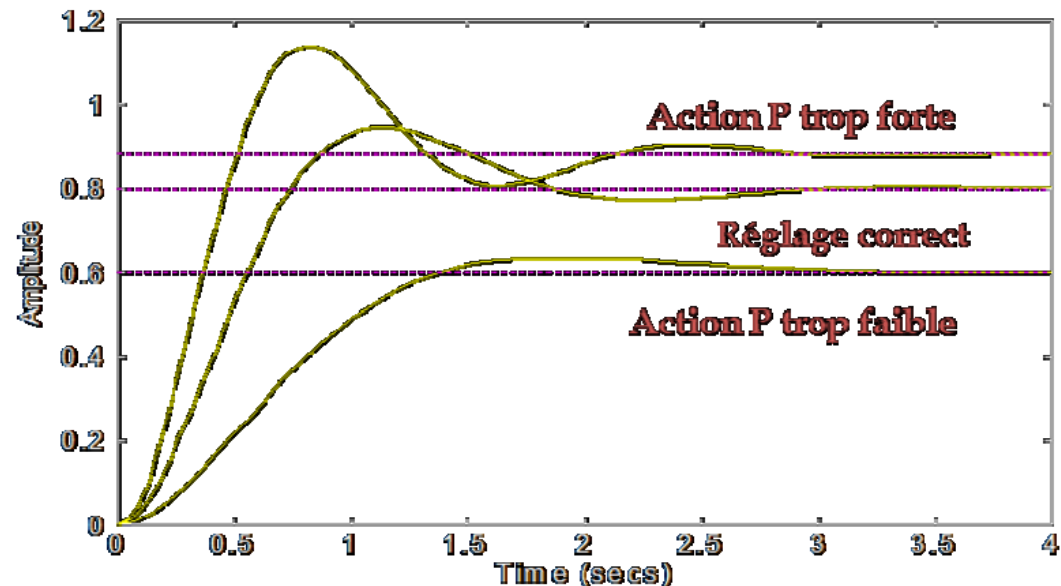
$$C(p) = \frac{u(p)}{\varepsilon(p)} = K_p$$

Caractéristiques:

- Améliore la rapidité,
- Précision imparfaite.
- Son augmentation peut causer l'instabilité du système.

Réglage:

On règle K_p telle que la réponse soit **la plus rapide**, avec un bon dépassement (ne dépassant pas 15 %)



3.2. Action intégrale (I):

Equation:

$$u(t) = \frac{1}{T_i} \int_0^t \varepsilon(\tau) d\tau$$

T_i : temps d'action intégrale ou le taux d'action intégrale.

T_i petit signifie une action intégrale élevée

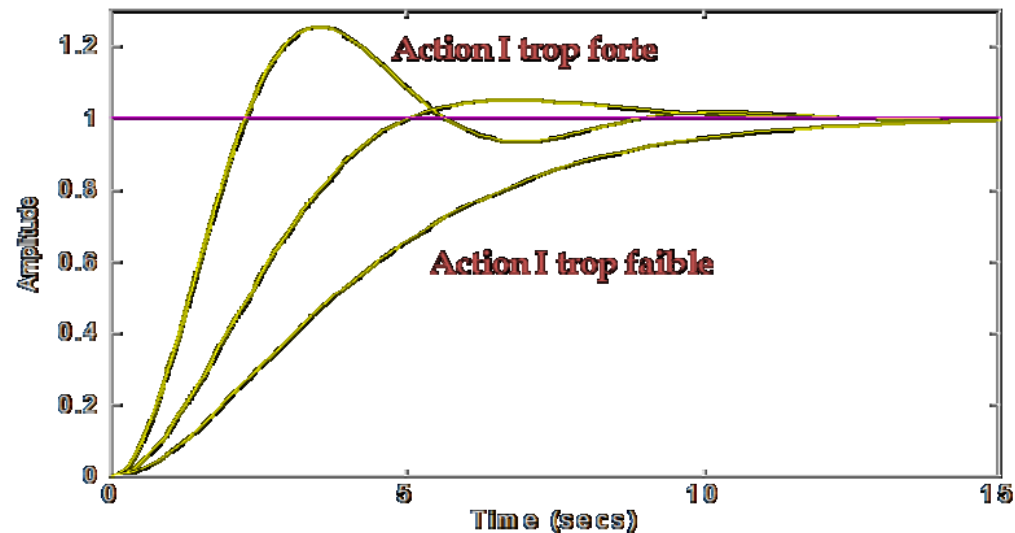
Caractéristiques:

- Améliore la précision,
- Filtre les bruits HF.
- Diminue la rapidité
- Diminue les marges de stabilité

Fonction de transfert:

$$C(p) = \frac{1}{T_i p}$$

Réglage:



3.3. Action dérivée (D):

Equation:

$$u(t) = T_d \frac{d\varepsilon(t)}{dt}$$

Fonction de transfert:

$$C(p) = T_d p \xrightarrow{\text{Version filtrée}} C(p) = \frac{T_d p}{1 + \frac{T_d}{N} p}$$

N'est pas propre $N \gg 1$

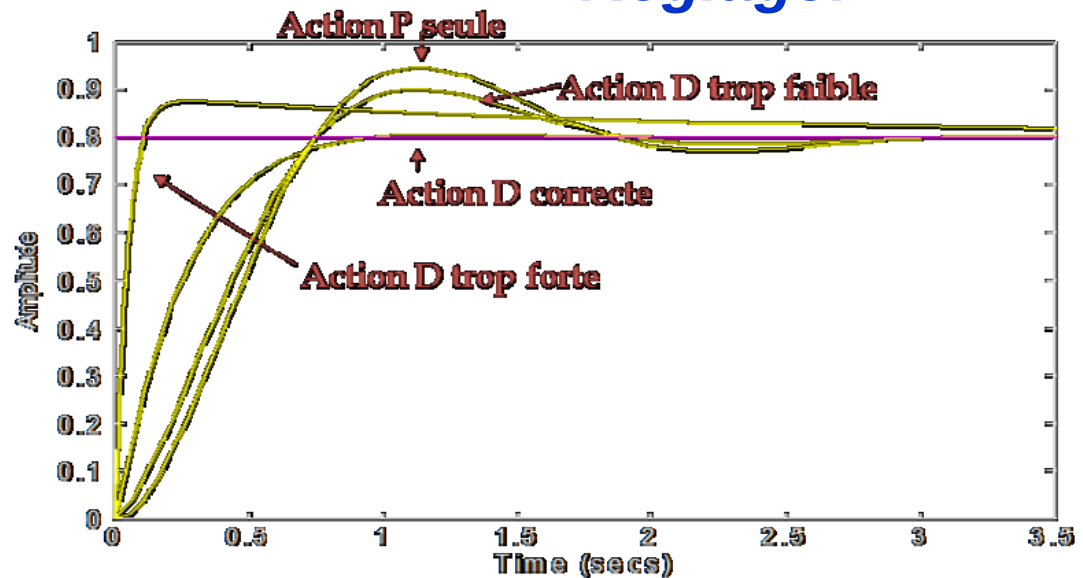
T_d : temps (ou taux) d'action dérivée.

T_d Grand signifie une action dérivée élevée

Caractéristiques:

- Améliore la stabilité ,
- Augmente la rapidité.
- Diminue la précision
- Amplifie les bruits HF

Réglage:



4. Le régulateur PI

Equation:

$$u(t) = K_p \varepsilon(t) + \frac{1}{T_i} \int_0^t \varepsilon(\tau) d\tau$$

Fonction de transfert:

Forme parallèle:

$$C(p) = K_p + \frac{1}{T_i p}$$

Forme série:

$$K \frac{(1 + \tau_i p)}{\tau_i p}$$

Caractéristiques:

- Annule l'erreur statique (contient une intégration), (action I)
- Améliore la rapidité, (action P)
- Augmente l'instabilité.

5. Le régulateur PID

Fonctions de transfert:

1

Forme
parallèle

$$C(p) = K_p + \frac{1}{T_i p} + T_d p$$

2

Forme
série

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

3

Forme
mixte

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

Caractéristiques:

- annule l'erreur statique,
- améliore la rapidité du système,
- améliore la stabilité du système.

C'est le correcteur le plus complet

5. Quelques méthodes de réglage des actions d'un PID industriel

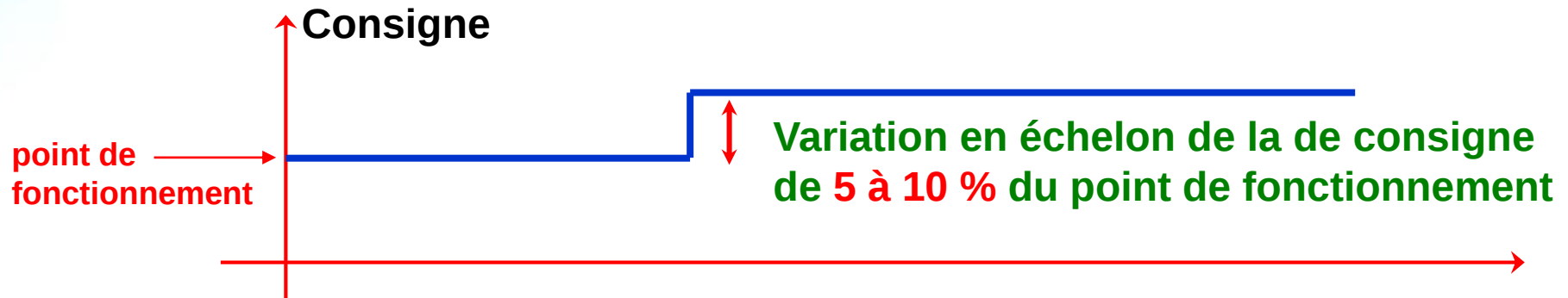
- Il existe **différentes méthodes** de réglage des actions d'un régulateur P.I.D.
- Suivant le type de procédé et les contraintes de fabrication on choisira l'une des méthodes:
 - Méthodes théorique: *connaissance du modèle et imposition d'un cahier des charges (marges stabilités, précision, rapidité, ...)*
 - Méthode par **approches successives**,
 - Méthode nécessitant l'identification du procédé,
 - Méthode de **Ziegler et Nichols**,
 - ...

5.1. Réglage par approches successives:

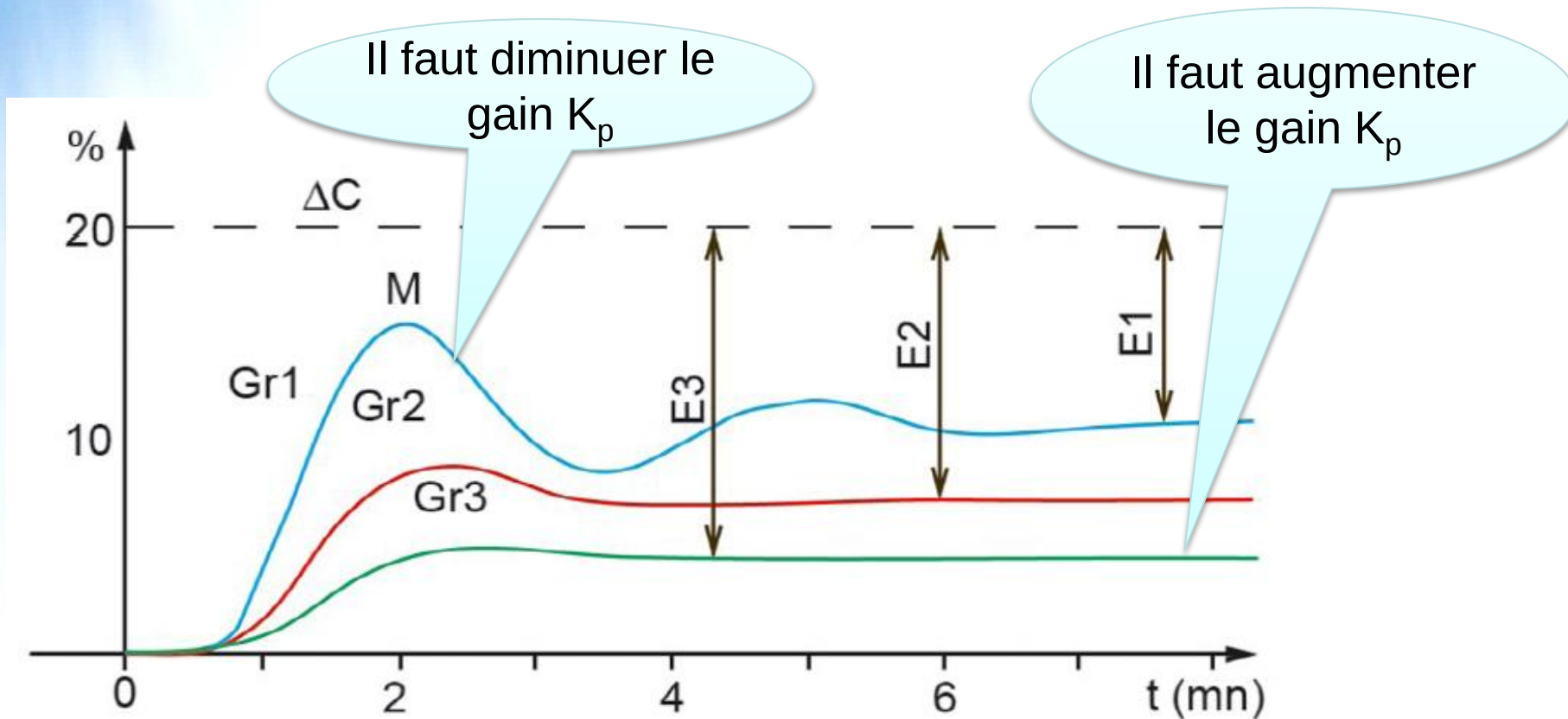
- Les actions seront réglées dans l'ordre **P**, **D**, **I**.
- Les critères de performance retenus pour la régulation sont :
 - ▶ une réponse bien amortie (*dépassement de 10 à 15 %*)
 - ▶ une rapidité maximum (*temps d'établissement minimal*).

Réglage de l'action proportionnelle (P):

- Stabiliser la mesure au point de fonctionnement.
- Mettre le régulateur en **P** seul, ($T_i = \text{max}$ et $T_d = 0$).
- Fixer un gain K_p faible ($K_p < 1$).
- Effectuer un échelon de consigne de **5 à 10 %**.
- Observer l'enregistrement de l'évolution du signal de mesure:



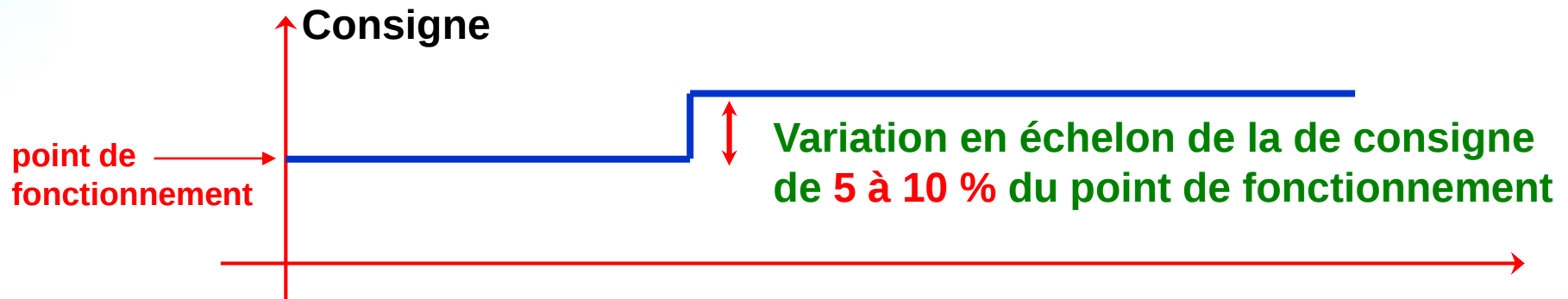
Réglage de l'action proportionnelle (P):



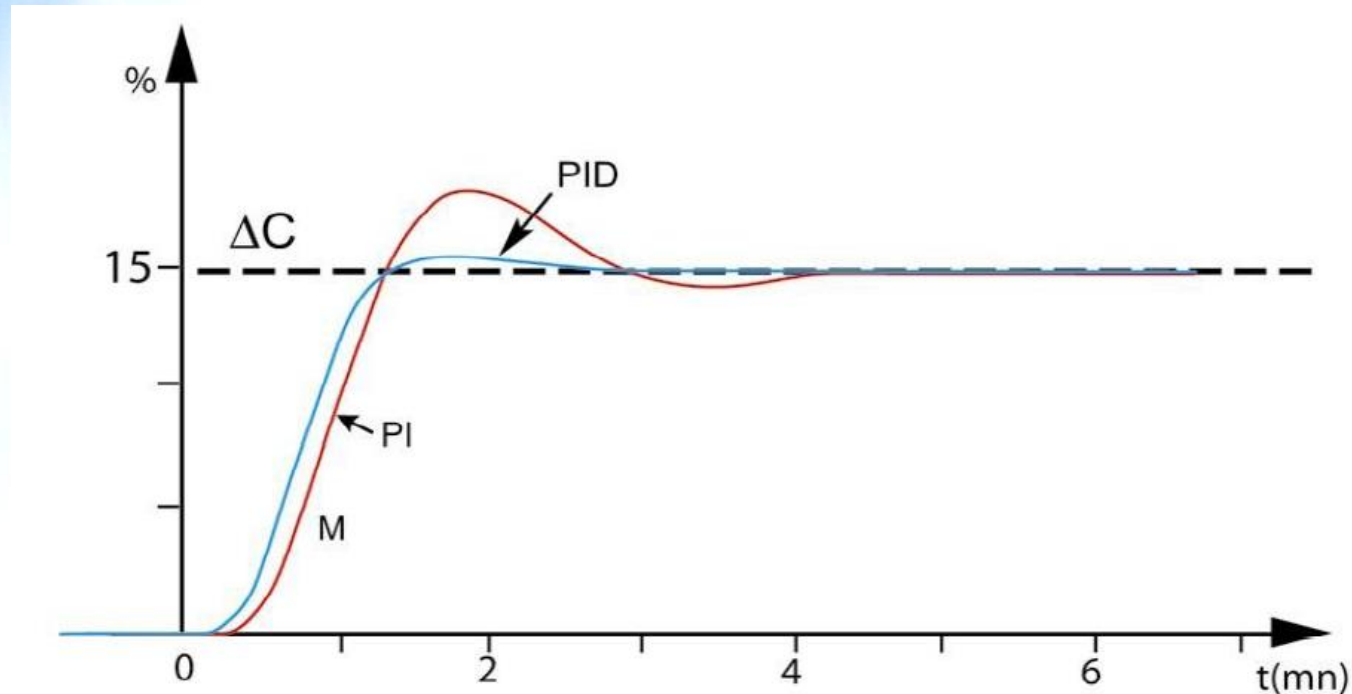
Il faut trouver un compromis entre rapidité et stabilité

Réglage de l'action Dérivée (D):

- L'action dérivée ne se justifie que si la mesure a un **certain retard**.
- Conserver la valeur de l'action proportionnelle déterminée précédemment et **l'intégrale minimale**.
- Fixer une action dérivée faible (T_d égal à quelques secondes ($tr/3$)).
- Effectuer un échelon de consigne de 5 à 10 % et observer la réponse:



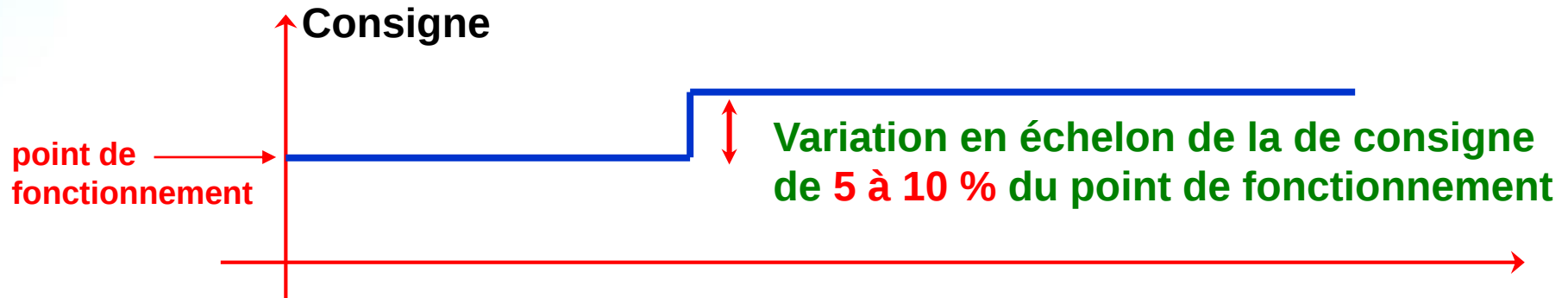
Réglage de l'action Dérivée (D):



- ▶ Si la réponse ne s'amortie pas, **diminuer T_d** .
- ▶ Si la réponse est plus lente, **augmenter T_d**

Réglage de l'action Intégrale (I):

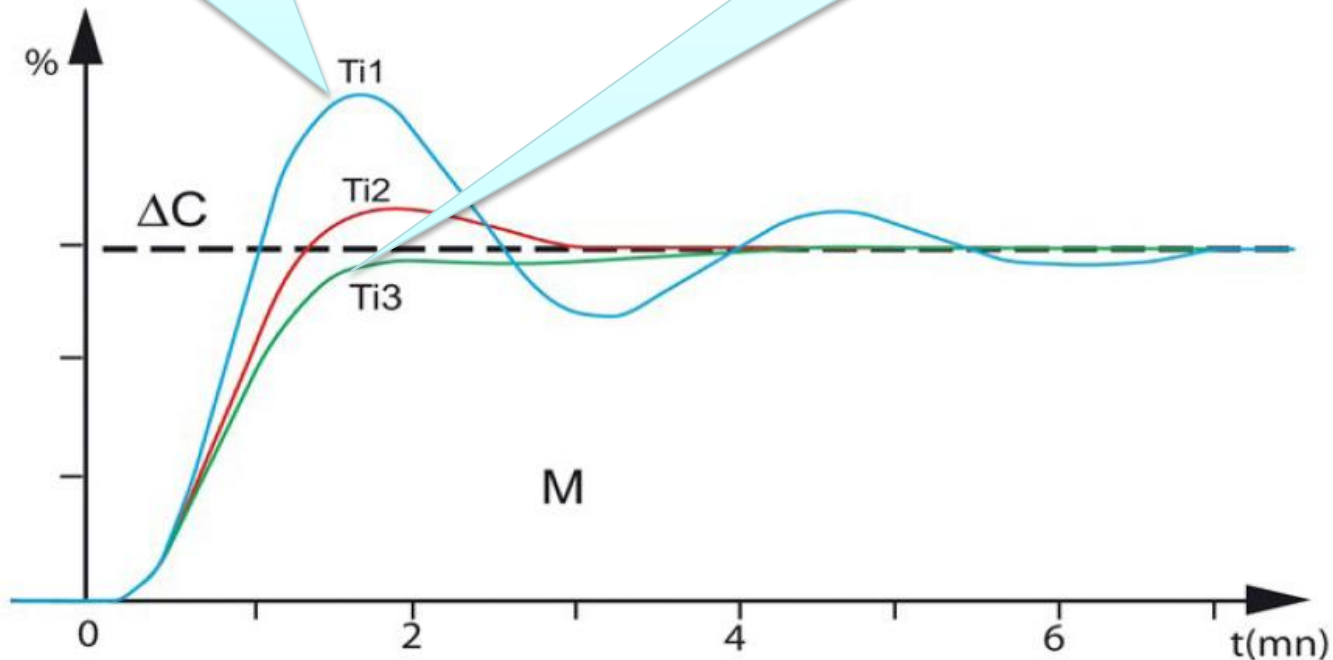
- Conserver les valeurs des actions proportionnelle et dérivée déterminées précédemment.
- Fixer une **action intégrale faible**.
- Effectuer un échelon de consigne de **5 à 10 %** et observer la réponse:



Réglage de l'action Intégrale (I):

Il faut augmenter T_i
(diminuer l'action I)

Il faut diminuer T_i
(augmenter l'action I)

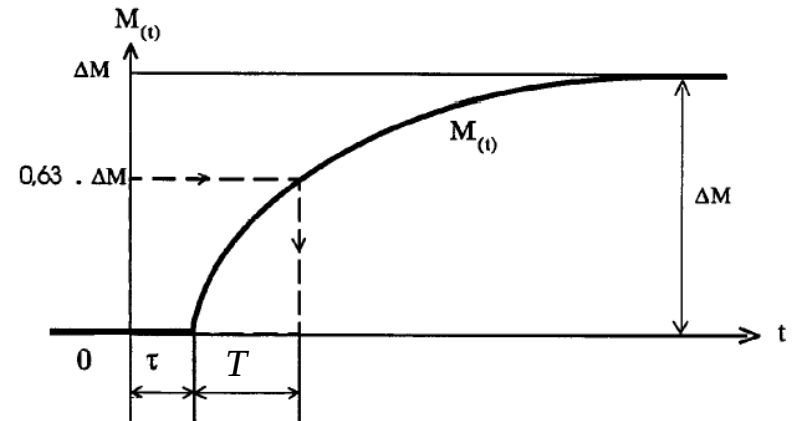
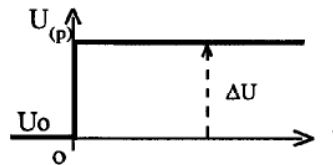


5.2. Réglage à partir de l'identification du procédé

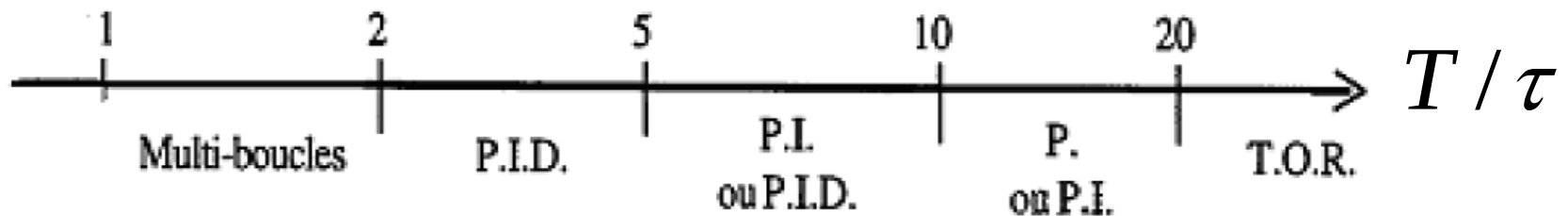
Cas d'un procédé stable:

Modèle de Brodia (1^{er} ordre = retard):

$$G(p) = \frac{Ke^{-\tau p}}{1 + Tp}$$



Le choix du mode de régulation est lié à la réglabilité du système déterminé par le rapport T / τ



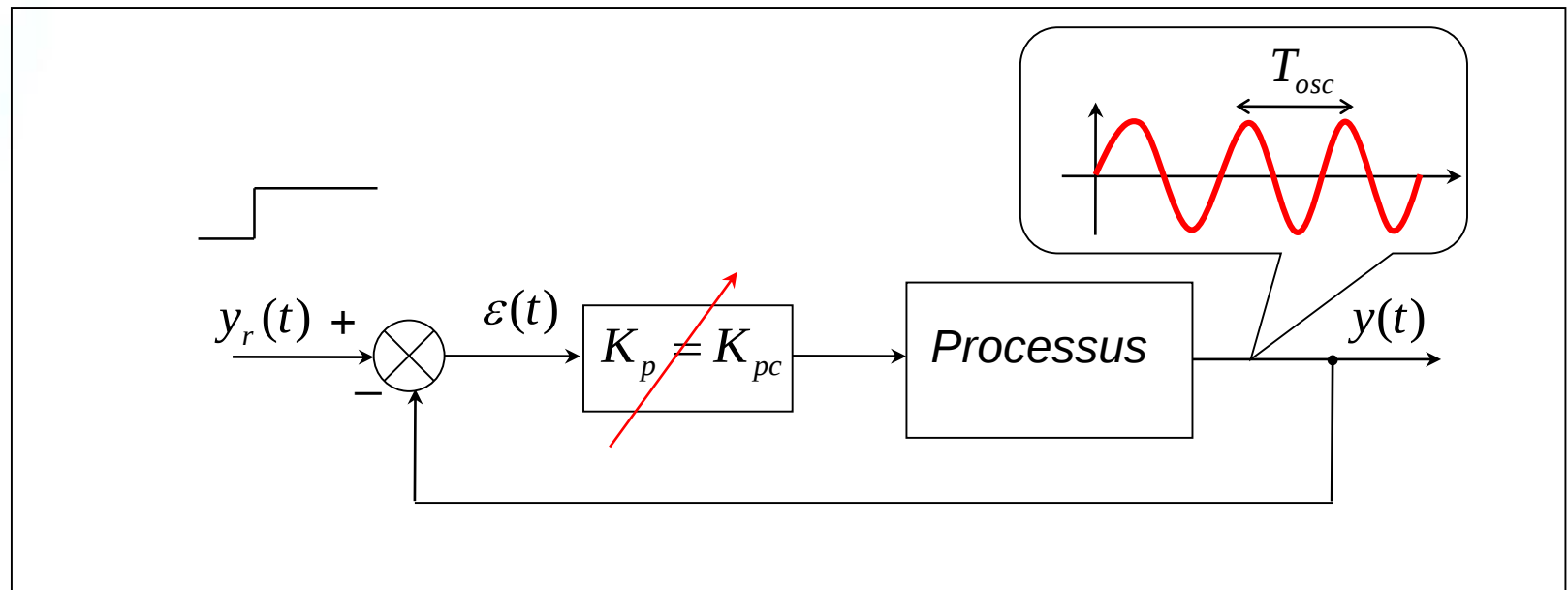
Cas d'un procédé stable:

Utiliser le **Tableau** suivant pour calculer les différentes actions:

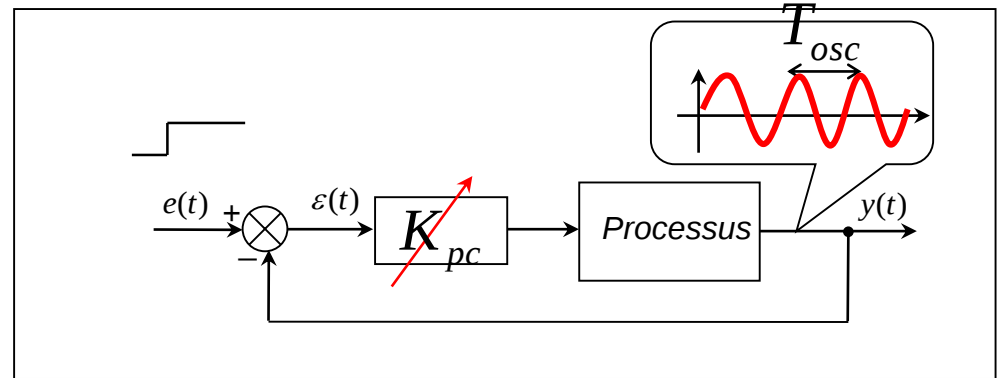
REGUL. ACTIONS	P	PI Série	PI Parallèle	PID série	PID Parallèle	PID Mixte
Kp (P)	$\frac{0.8 \times T}{K \times \tau}$	$\frac{0.8 \times T}{K \times \tau}$	$\frac{0.8 \times T}{K \times \tau}$	$\frac{0.85 \times T}{K \times \tau}$	$\frac{(T/\tau) + 0.4}{1.2 \times K}$	$\frac{(T/\tau) + 0.4}{1.2 \times K}$
Ti (1/I)	Maxi (I=0)	T	$\frac{K \times \tau}{0.8}$	T	$\frac{K \times \tau}{0.75}$	$T + 0.4 \times \tau$
Td (D)	0	0	0	$0.4 \times \tau$	$\frac{0.35 \times T}{K}$	$\frac{T \times \tau}{\tau + 2.5 \times T}$

5.3. Réglage des régulateurs par la méthode de Zeigler Nichols

- Méthode utilisable si le système étudié **supporte les dépassements**,
- utilisée quand le **processus est instable** en BO, ou qu'il **n'est pas techniquement possible d'ouvrir la boucle**.
- On réalise alors un **test de pompage**:



Réglage:



- on règle :

Le gain proportionnel jusqu'au gain critique K_{pc} ,

- On annule l'action intégrale : $T_i = \infty$ ($I=0$),
- On annule l'action dérivée : $T_d = 0$ ($D=0$),
- On mesure alors la période des oscillations, T_{osc} , que fait la sortie du système.

A partir des valeurs de K_{pc} et T_{osc} on détermine les paramètres des régulateurs standards (utiliser le tableau suivant:)

Modes de régul. Actions	P	PI Série	PI Parallèle	PID série	PID Parallèle	PID Mixte
K_p	$\frac{K_{pc}}{2}$	$\frac{K_{pc}}{2.2}$	$\frac{K_{pc}}{2.2}$	$\frac{K_{pc}}{3.3}$	$\frac{K_{pc}}{1.7}$	$\frac{K_{pc}}{1.7}$
T_i	∞	$\frac{T_{osc}}{1.2}$	$\frac{2T_{osc}}{K_{pc}}$	$\frac{T_{osc}}{4}$	$\frac{0.85T_{osc}}{K_{pc}}$	$\frac{T_{osc}}{2}$
T_d	0	0	0	$\frac{T_{osc}}{4}$	$\frac{K_{pc}T_{osc}}{13.3}$	$\frac{T_{osc}}{8}$

6.5. Exemple: Régulation de vitesse d'un MCC à l'aide de la carte Arduino



```
#include <PID_v1.h>
```

← Utiliser la bibliothèque **PID_v1.h**

```
// Define PID constants
```

```
double Kp = 1.0;
```

```
double Ki = 0.1;
```

```
double Kd = 0.1;
```

```
// Define PID variables
```

```
double setpoint = 100; // Consigne de vitesse
```

```
double input, output;
```

```
PID myPID(&input, &output, &setpoint, Kp, Ki, Kd, DIRECT);
```

créer un objet de contrôleur PID
en utilisant la bibliothèque
PID_v1.h.

DIRECT: le contrôleur agira pour
augmenter la sortie lorsque
l'erreur est positive

```
void setup() {
```

```
  // Initialize PID
```

```
  myPID.SetMode(AUTOMATIC);
```

← **AUTOMAIC**: le contrôleur en BF. Il existe aussi **MANUAL**
(en BO) et **PAUSE**

```
  myPID.SetSampleTime(100); // Temps d'échantillonnage en millisecondes
```

```
}
```

```
void loop() {
```

```
  // Mesurez la vitesse actuelle du moteur (supposez qu'elle est stockée dans input)
```

```
  input = analogRead(A0); // Exemple de lecture analogique de la vitesse du moteur
```

```
  // Calculez la sortie du PID
```

```
  myPID.Compute();
```

← calcule la sortie du contrôleur PID en fonction des valeurs
actuelles de l'erreur, de la consigne et des coefficients PID

```
  // Appliquez la sortie au Hacheur qui commande le moteur (par exemple, en utilisant la sortie PWM)
```

```
  analogWrite(3, output); // Supposons que le moteur est connecté à la broche PWM 3
```

```
  delay(100); // Attendez pour le prochain échantillonnage
```

```
}
```

7. Exemples de PID industriels

PID industriels conventionnels:

ABB : CM15, CM30, CM50



Honeywell : série UDC1200, UDC1700

Eurotherm : série 3200



Omron : série E5CC



PID industriels intégrés dans des Automates:

ABB : AC500

Schneider Electric : Modicon M241

Siemens : S7-1200



Exemple de régulateur: **P.M.A PID**
Temperature Controller: KS 90-1 / 92-1

<https://docs.rs-online.com/9aee/0900766b81259404.pdf>

face avant

Boutons pour le paramétrage
manuel



- 2 Entrées Universelles (la 3^{ème} en option)
- 1 entrée analogique
- Régulation PID
- Échantillonnage 100 ms
- RS 422/485 Modbus RTU interface
- PROFIBUS-DP interface
- 4 Sorties en standard (4 relais, 3 relais + 1 analogique, 2 relais+2 analogiques), Alim Transmetteur, Sorties digitales supplémentaires,
- Logiciel de configuration/paramétrage (connexion en face avant)
-

