

## Régulation PID des processus à base d'automates SIMATIC S7

### 1. Principe de régulation

La grandeur à réguler est mesurée de façon continue et est ensuite comparée avec une valeur consigne donnée en entrée du système. La régulation devra, en fonction du résultat de la comparaison, rapprocher la valeur à réguler de la consigne

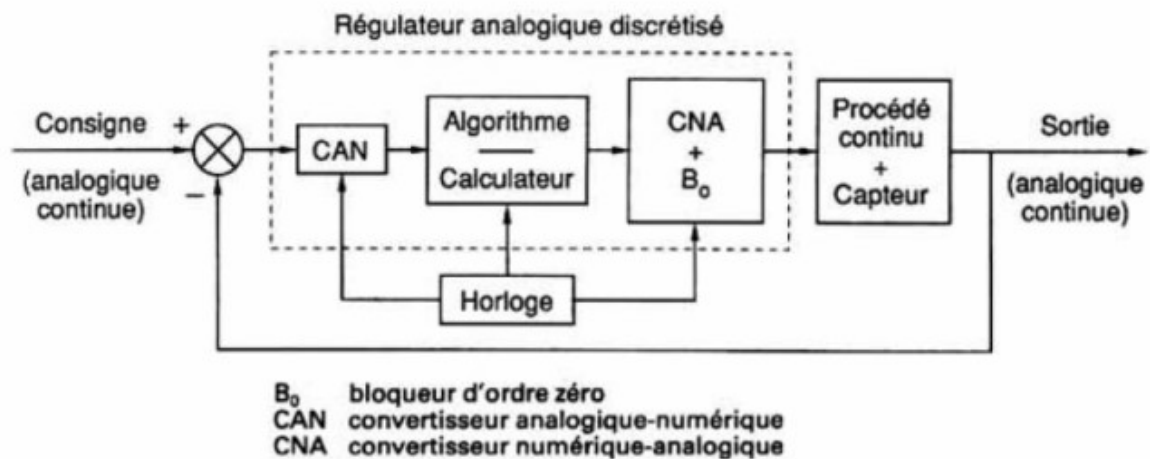


Figure 1 : Composition d'une boucle de régulation numérique

Le régulateur est l'élément central d'un asservissement. Il évalue l'erreur calculée par le comparateur, c'est à dire l'écart entre la sortie et la consigne. Il en déduit une valeur de correction à transmettre au procédé afin de corriger la sortie.

La manière (l'algorithme) dont le régulateur calcule la valeur de correction (Commande) à partir de l'erreur est la principale activité de l'asservissement.

### 2. Concept de la fonction de régulation PID dans SIMATIC S7

La fonction « Régulation PID » se compose des éléments suivants :

- Les blocs fonctionnels avec deux variantes :
  - **CONT\_C continu** : pour un signal de commande continue. Ce bloc sera détaillé par la suite.
  - **PULSGEN impulsion** : pour un signal de commande modulé en largeur d'impulsion.
- L'interface de paramétrage pour la configuration des blocs fonctionnels.

On peut utiliser le régulateur PID comme régulateur de maintien autonome mais aussi comme régulateur en cascade, de mélange ou de rapport dans des régulations à plusieurs boucles. Sa méthode de travail se base sur l'algorithme PID du régulateur à échantillonnage à sortie

analogique, complété le cas échéant par un étage conformateur d'impulsions assurant la formation des signaux de sortie modulés en largeur d'impulsions

### 3. Le bloc FB 41 « CONT\_C »

Cette variante du régulateur PID, comme déjà précisé, utilisé en tant que régulateur continu. Son rôle sera de calculer une valeur d'ajustement  $Y_r$  en fonction de l'erreur (différence mesure/consigne)  $e = w - x$ , selon l'algorithme d'un régulateur P.I.D, et de livrer cette grandeur d'ajustement  $y$  sur sa sortie analogique.

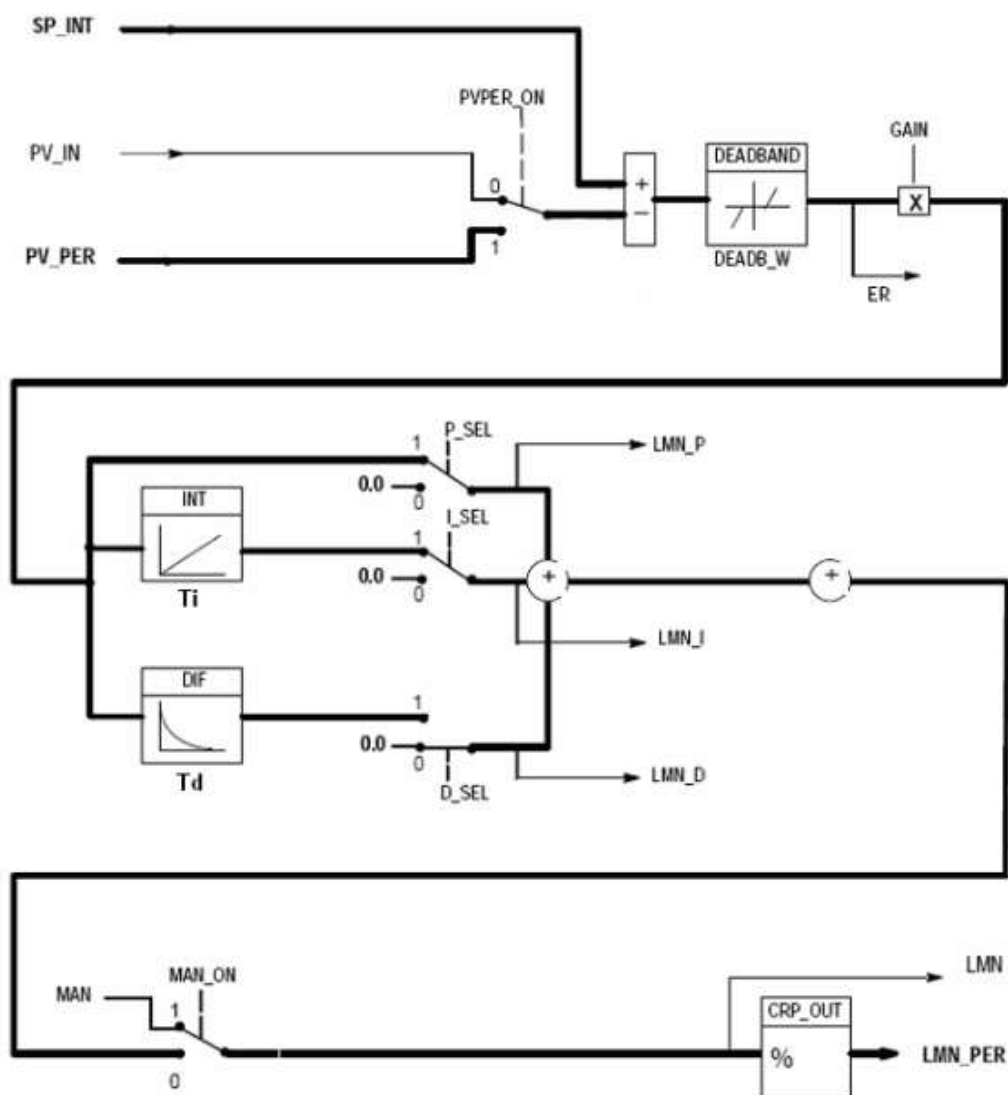


Figure 2 : L'algorithme du bloc FB41

Le FB 41 propose en plus d'une régulation purement logicielle, des fonctions de traitement de la consigne, de la mesure et la grandeur de réglage calculée.

Ainsi la configuration du bloc, nécessite l'ajustement d'un grand nombre de paramètres et de sous-fonctions, dont nous présentons ci-dessous le détail :

• **Les paramètres et les grandeurs d'entrées :**

<b>Paramètres d'entées</b>	<b>Types de données</b>	<b>Description</b>
MAN_ON	BOOL	<b>MANUAL VALUE ON</b> / Activation du mode manuel Quand cette entrée est à 1, la boucle de régulation est interrompue. La valeur de réglage manuelle est sortie comme grandeur de réglage.
PVPER_ON	BOOL	<b>PROCESS VARIABLE PERIPHERY ON</b> / Activation de la mesure de périphérie Pour que la mesure soit lue en périphérie, il faut relier l'entrée PV_PER à la périphérie et mettre à 1 l'entrée PVPER_ON.
P_SEL	BOOL	<b>PROPORTIONAL ACTION ON</b> / Activation de l'action proportionnelle
I_SEL	BOOL	<b>INTEGRAL ACTION ON</b> / Activation de l'action par intégration
D_SEL	BOOL	<b>DERIVATIVE ACTION ON</b> / Activation de l'action par dérivation
CYCLE	TEMPS	<b>SAMPLE TIME</b> / Période d'échantillonnage Le temps s'écoulant entre les appels de bloc doit être constant. Il est indiqué au niveau de cette entrée.
SP_INT	REEL	<b>INTERNAL SETPOINT</b> / Consigne interne Cette entrée sert à introduire une valeur de consigne.
PV_IN	REEL	<b>PROCESS VARIABLE IN</b> / Mesure d'entrée. Cette entrée permet de paramétrer une valeur de mise en service ou d'appliquer une mesure externe en virgule flottante.
PV_PER	MOT	<b>PROCESS VARIABLE PERIPHERIE</b> / Mesure de périphérie. La mesure en format de périphérie est appliquée au régulateur par cette entrée.
MAN	REEL	<b>MANUAL VALUE</b> / Valeur de réglage manuelle Cette entrée sert à introduire une valeur de réglage manuelle moyennant des fonctions de contrôle-commande.
GAIN	REEL	<b>PROPORTIONAL GAIN</b> / Coefficient d'action proportionnelle
TI	TEMPS	<b>RESET TIME</b> / Temps d'intégration..
TD	TEMPS	<b>DERIVATIVE TIME</b> / Temps de dérivation.
DEADB_W	REEL	<b>DEAD BAND WIDTH</b> / Largeur de zone morte. Le signal d'erreur traverse une zone morte. Cette entrée détermine la taille de la zone morte pour l'erreur de réglage.

**Tableau 1 :** Les paramètres d'entrée du bloc fonctionnel FB 41

• Les paramètres de sorties

Paramètres de sorties	Types de données	Description
LMN	REEL	<i>MANIPULATED VALUE</i> / Valeur de réglage.  Cette sortie donne en virgule flottante la valeur de réglage agissant réellement.
LMN_PER	MOT	<i>MANIPULATED VALUE PERIPHERY</i> / Valeur de réglage de périphérie.  Cette sortie fournit la valeur de réglage en format de périphérie.
LMN_P	REEL	<i>PROPORTIONALITY COMPONENT</i> / Composante P.  Cette sortie contient la composante proportionnelle de la grandeur de réglage.
LMN_I	REEL	<i>INTEGRAL COMPONENT</i> / Composante I. Cette sortie contient la composante intégrale de la grandeur de réglage.
LMN_D	REEL	<i>DERIVATIVE COMPONENT</i> / Composante D.  Cette sortie contient la composante différentielle de la grandeur de réglage.

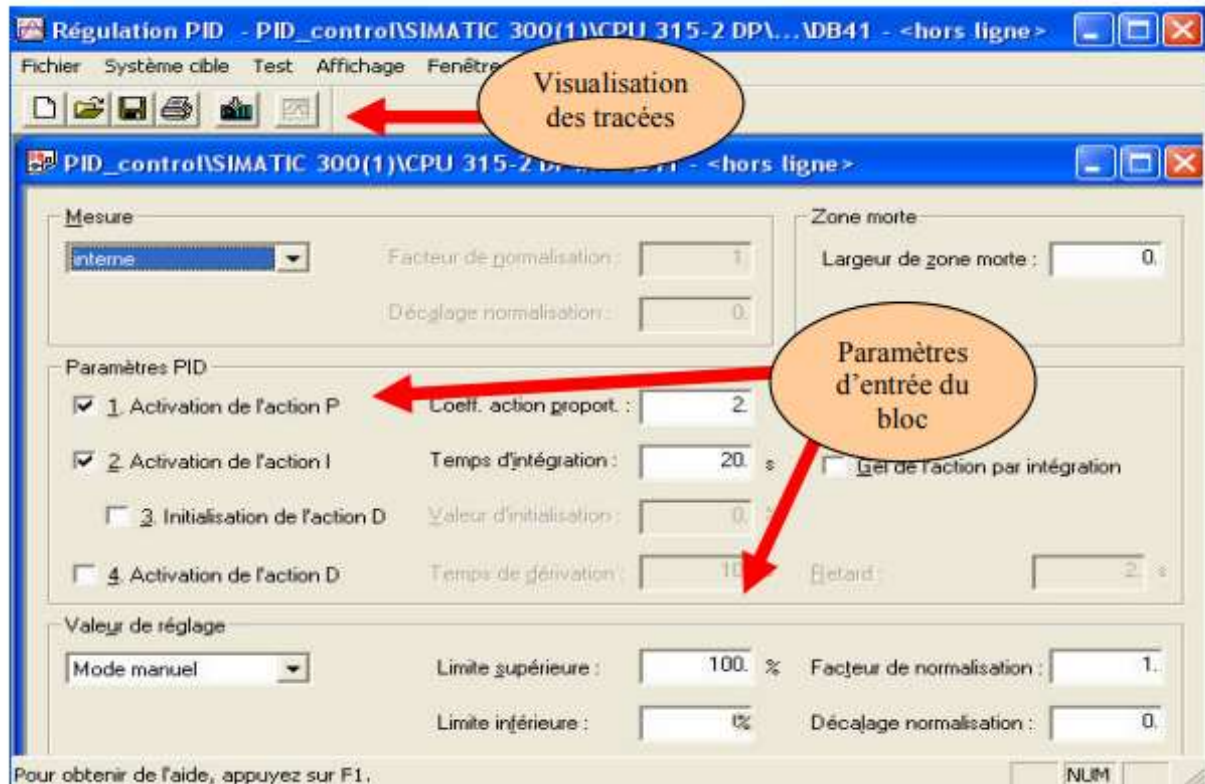
**Tableau 2 :** Les paramètres de sortie du bloc fonctionnel FB 41

Pour une CPU donnée, il faut trouver un compromis entre le nombre de régulateurs et la fréquence de traitement de chacun d'eux. Plus les boucles de régulation seront rapides, c'est-à-dire plus souvent les grandeurs réglantes sont à calculer par unité de temps, plus faible sera le nombre de régulateurs.

Il n'y a pas de restriction concernant le type de processus à mettre en œuvre. Les blocs fonctionnels s'appliquent aux systèmes inertiels (températures, niveaux de remplissages, etc.) comme aux systèmes très rapides (débit, vitesse de rotation, etc.).

#### 4. Paramétrage du régulateur PID du type continue FB 41 avec STEP 7

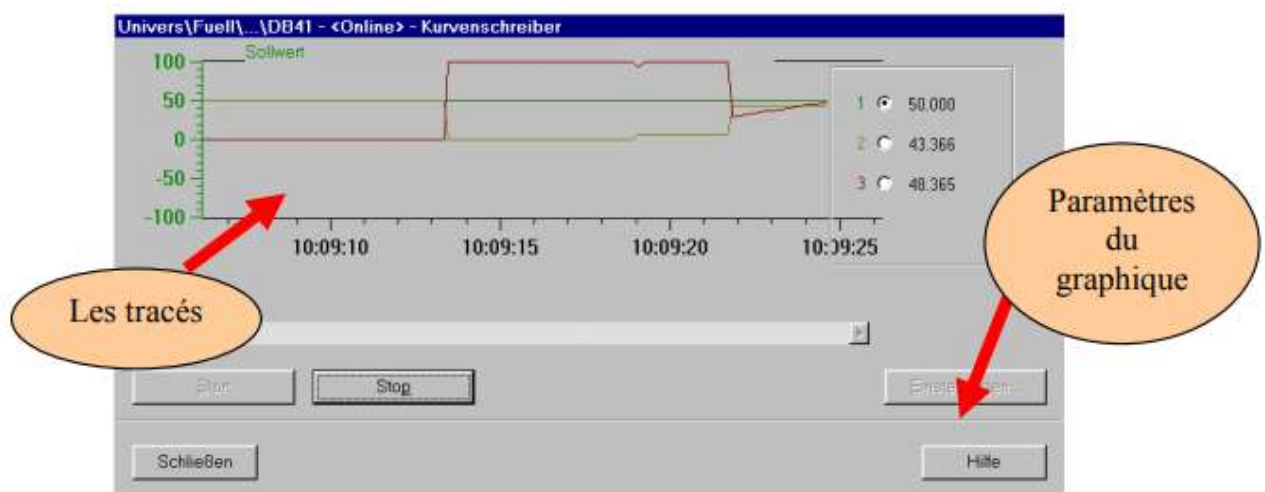
Pour l'utilisation et l'optimisation du fonctionnement du régulateur PID, on fait appel à l'outil « Paramétrage de la régulation PID », les paramètres choisis seront sauvegardés dans le DB d'instance local associé au bloc FB41.



**Figure 3 : Paramétrage du régulateur PID**

Une fois le bloc de données paramétré, sauvegardé et chargé dans la CPU, c'est alors que peut intervenir la mise en route de la représentation graphique pour observer le comportement de la régulation.

Les courbes de la valeur souhaitée (consigne), la valeur réelle (mesure) et la grandeur d'ajustement (commande) peuvent être visualisées en temps réel grâce au graphique.



**Figure 4 : Tracée des trois courbe consigne-mesure-commande**

## 5. Réglage de la commande PID :

Le choix d'un régulateur pour un système en boucle fermée dépend des performances que l'on souhaite obtenir pour ce système. Les performances souhaitées sont généralement :

- le dépassement réduit
- le temps de réponse rapide ou la bande passante large
- l'écart entre la sortie et la consigne du système est nul en régime permanent.
- une bonne robustesse par rapport à la perturbation et aux variations des paramètres du système.

Il existe plusieurs méthodes pour calculer les paramètres du régulateur, pour que le système en boucle fermée ait la réponse désirée. Ces méthodes peuvent être empiriques ou analytiques.

### 5.1 Méthodes de Ziegler et Nichols (ZN)

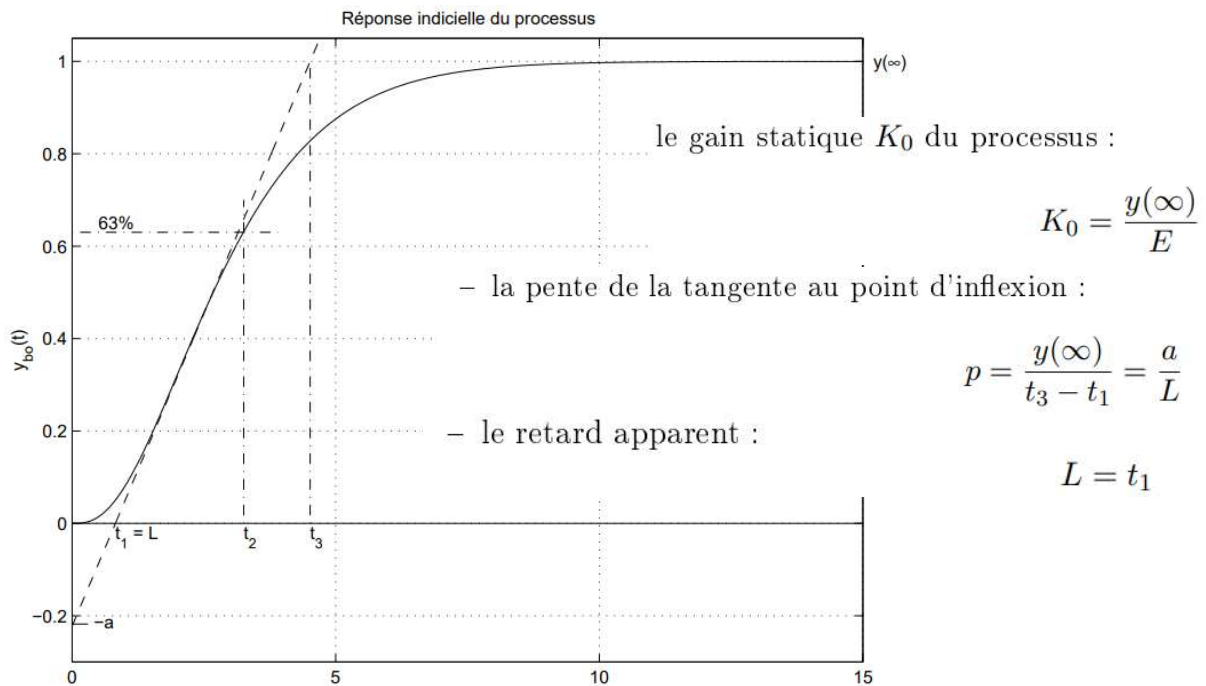
En 1942, Ziegler et Nichols ont proposé deux approches heuristiques basées sur leur expérience et quelques simulations pour ajuster rapidement les paramètres des régulateurs P, PI et PID. La première méthode nécessite l'enregistrement de la réponse indicielle en boucle ouverte, alors que la deuxième demande d'amener le système bouclé à sa limite de stabilité.

Pour obtenir les paramètres du régulateur PID, il suffit d'enregistrer la réponse indicielle du processus seul (c'est-à-dire sans le régulateur), puis de tracer la tangente au point d'inflexion de la courbe. On mesure ensuite sa pente  $p$ , le retard apparent  $L$  correspondant au point d'intersection de la tangente avec l'abscisse et le gain  $K_0 = y_\infty/E$  (figure 5). On peut alors calculer les coefficients du régulateur choisi à l'aide du tableau 1.

Généralement, les gains  $K_p$  proposés par Ziegler-Nichols sont trop élevés et conduisent à un dépassement supérieur à 20%. Il ne faut donc pas craindre de réduire  $K_p$  d'un facteur 2 pour obtenir une réponse satisfaisante.

Type	$K_p$	$T_i$	$T_d$
P	$1/(pLK_0) = 1/(aK_0)$		
PI	$0.9/(pLK_0) = 0.9/(aK_0)$	$3L$	
PID	$1.2/(pLK_0) = 1.2/(aK_0)$	$2L$	$0.5L$

**Tableau 3 :** Paramètres PID obtenus à partir d'une réponse indicielle (ZN)



**Figure 5 :** Réponse indicielle d'un processus

## 5.2 Méthode de CHIEN-HRONES RESWICK

Cette méthode s'applique au système ayant une fonction de transfert du premier ordre qui a une constante de temps  $T$  avec retard  $\tau$ . Le réglage des paramètres se fait soit par rapport à la consigne (poursuite), soit par rapport à la perturbation (régulation). Deux réglages de cette méthode sont proposés (voir tableau ci-dessous) pour des dépassements ( $D$ ) respectifs de 0% et 20% avec un temps de montée ( $t_r$ ) minimum.

Correcteur	$D = 0\%$ $t_r$ minimum		$D = 20\%$ $t_r$ minimum	
	Régulation	Poursuite	Régulation	Poursuite
$k_0$	$0,95 \frac{T}{\tau}$	$0,6 \frac{T}{\tau}$	$1,2 \frac{T}{\tau}$	$0,95 \frac{T}{\tau}$
$T_i$	$2,4\tau$	$T$	$2\tau$	$1,35T$
$T_d$	$0,42\tau$	$0,5\tau$	$0,42\tau$	$0,47\tau$

$k_0 = KK_p$  représente le gain de la boucle ouverte ( $K, K_p$  respectivement le gain du procédé et le gain du régulateur). Le problème avec cette méthode est qu'il faut choisir un jeu de paramètres ( $k_0, T_i, T_d$ ), soit en régulation soit en poursuite, donc les performances de la boucle fermée avantageront l'un par rapport à l'autre en fonction du choix des paramètres ( $k_0, T_i, T_d$ ).