
Présentation

Objet de ce chapitre

Ce chapitre présente les **fonctions de régulation de base** du logiciel PL7.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Présentation générale	346
Principe de la boucle de régulation	347
Méthodologie de développement d'une application de régulation	348

Présentation générale

Généralités

Les fonctions de régulation sont des **éléments de base** du langage PL7. Elles permettent de programmer des boucles de régulation sur automates Micro et Premium.

Ces fonctions sont particulièrement adaptées pour :

- répondre aux besoins de process séquentiel nécessitant des fonctions de régulation auxiliaire (exemples : machines d'emballage à film plastique, machines de traitement de surface, presses...),
- répondre aux besoins des process de régulation simple (exemples : fours de traitements de métaux, fours à céramiques, petits groupes frigorifiques...),
- répondre à des particularités d'asservissement ou de régulation mécanique dont le temps d'échantillonnage est critique (exemples: régulation de couple, régulation de vitesse).

Un interfaçage préconfiguré avec la gamme des CCX_17 permet le pilotage et le réglage des boucles de régulation. Dans ce cadre, jusqu'à 9 boucles de régulation sont accessibles par le CCX_17.

<p>Note : Il n'y a pas de limitation du nombre de fonctions PID dans une application. En pratique, c'est le nombre maximal de modules d'entrées et de sorties accepté par l'automate qui limite le nombre de boucles.</p>
--

Fonctions disponibles

Les fonctions de régulation de base se répartissent en deux catégories :

- une famille de fonctions algorithmiques :
 - fonction PID pour réaliser une correction de type PID mixte (série - parallèle),
 - fonction PWM pour réaliser les adaptations de modulation en durée sur sorties TOR,
 - fonction SERVO pour réaliser les adaptations de commande de moteur,
- une fonction de dialogue opérateur (PID_MMI) qui intègre un applicatif de pilotage et de réglage des PID de l'application sur CCX_17 version 2.

La fonction PID_MMI est associée à 3 types d'écrans préconfigurés.

Principe de la boucle de régulation

Présentation

Le fonctionnement d'une boucle de régulation comprend trois phases distinctes :

- l'acquisition des données :
 - mesure(s) provenant des capteurs du process (analogiques, codeurs),
 - consigne(s) provenant généralement de variables internes de l'automate ou de données issues du CCX_17.
- l'exécution de l'algorithme de régulation PID,
- l'envoi des commandes adaptées aux caractéristiques des actionneurs à piloter via des sorties TOR ou analogiques.

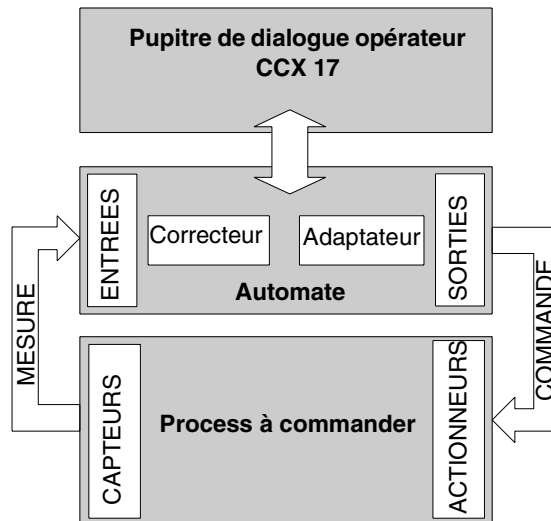
L'algorithme PID élabore le signal de commande à partir :

- de la mesure échantillonnée par le module d'entrée,
- de la valeur de la consigne fixée soit par l'opérateur, soit par programme,
- des valeurs des différents paramètres du correcteur.

Le signal issu du correcteur est soit traité directement par une carte de sortie analogique de l'automate raccordé à l'actionneur, soit traité via les adaptations PWM ou SERVO en fonction des types d'actionneur à piloter sur une carte de sortie TOR de l'automate.

Illustration

L'illustration ci-dessous schématise le principe d'une boucle de régulation.

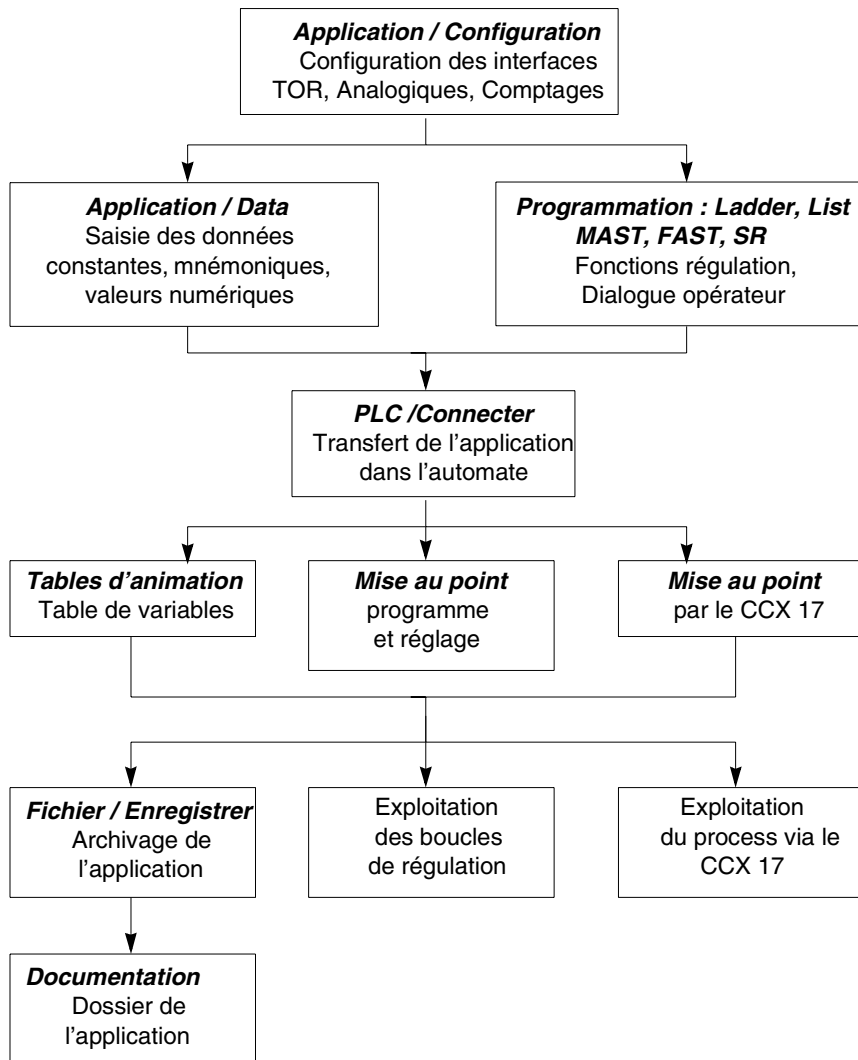


Méthodologie de développement d'une application de régulation

Schéma de principe

Le schéma ci-dessous décrit l'enchaînement des tâches à effectuer lors de la création et la mise au point d'une application de régulation.

Note : L'ordre défini est donné à titre indicatif.



Description des fonctions de régulation

30

Présentation

Objet de ce chapitre

Ce chapitre décrit les fonctions de régulation.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Programmation d'une fonction de régulation	350
Fonction PID	351
Programmation de la fonction PID	353
Fonction PWM	357
Programmation de la fonction PWM	359
Fonction SERVO	361
Programmation de la fonction SERVO	364
Comportement des fonctions dans les modes de marche	367

Programmation d'une fonction de régulation

Règles de programmation

Les paramètres des fonctions régulation doivent obligatoirement tous être renseignés. Les fonctions utilisent trois types de paramètres :

- des paramètres en lecture seule, pris en compte en début d'exécution de la fonction,
- des paramètres en écriture seule, positionnés à l'issue de l'exécution de la fonction,
- des paramètres en lecture et en écriture, dont les contenus sont pris en compte au début de l'exécution de la fonction et sont ensuite remis à jour par les résultats de la fonction.

Note : Les fonctions de régulation doivent être programmées dans une tâche **périodique** (MAST périodique ou FAST). Elles ne doivent pas être conditionnées.

Paramétrage

Les paramètres d'entrée de type mot sont des grandeurs analogiques exprimées dans l'échelle [0, +10000] et peuvent être directement connectés aux capteurs de mesure via les mots %IWxy.i des entrées analogiques.

Les paramètres de sortie de type bit permettent de commander des actionneurs de type TOR et peuvent être directement connectés à des variables de type %Qxy.i.

De la même façon, les paramètres de sortie de type mot permettent de commander des actionneurs de type analogique sur l'échelle [0, +10000] et peuvent être directement affectés à des variables de type %QWxy.i.

Les paramètres de type tables de mots %MWi:L regroupent des paramètres utilisateurs et les données nécessaires au fonctionnement interne de la fonction.

Si la longueur d'une table est insuffisante, la fonction ne s'exécute pas.

Note : Afin de conserver les paramètres de réglage des OF régulation sur démarrage à froid, il est nécessaire de supprimer l'option de remise à zéro des %MWi (dans écran de configuration du processeur)

Fonction PID

Généralités

La fonction PID réalise une correction PID à partir d'une mesure et d'une consigne analogique au format [0 - 10000] et fournit une commande analogique au même format.

Fonctions disponibles

L'OF PID comporte les fonctions suivantes :

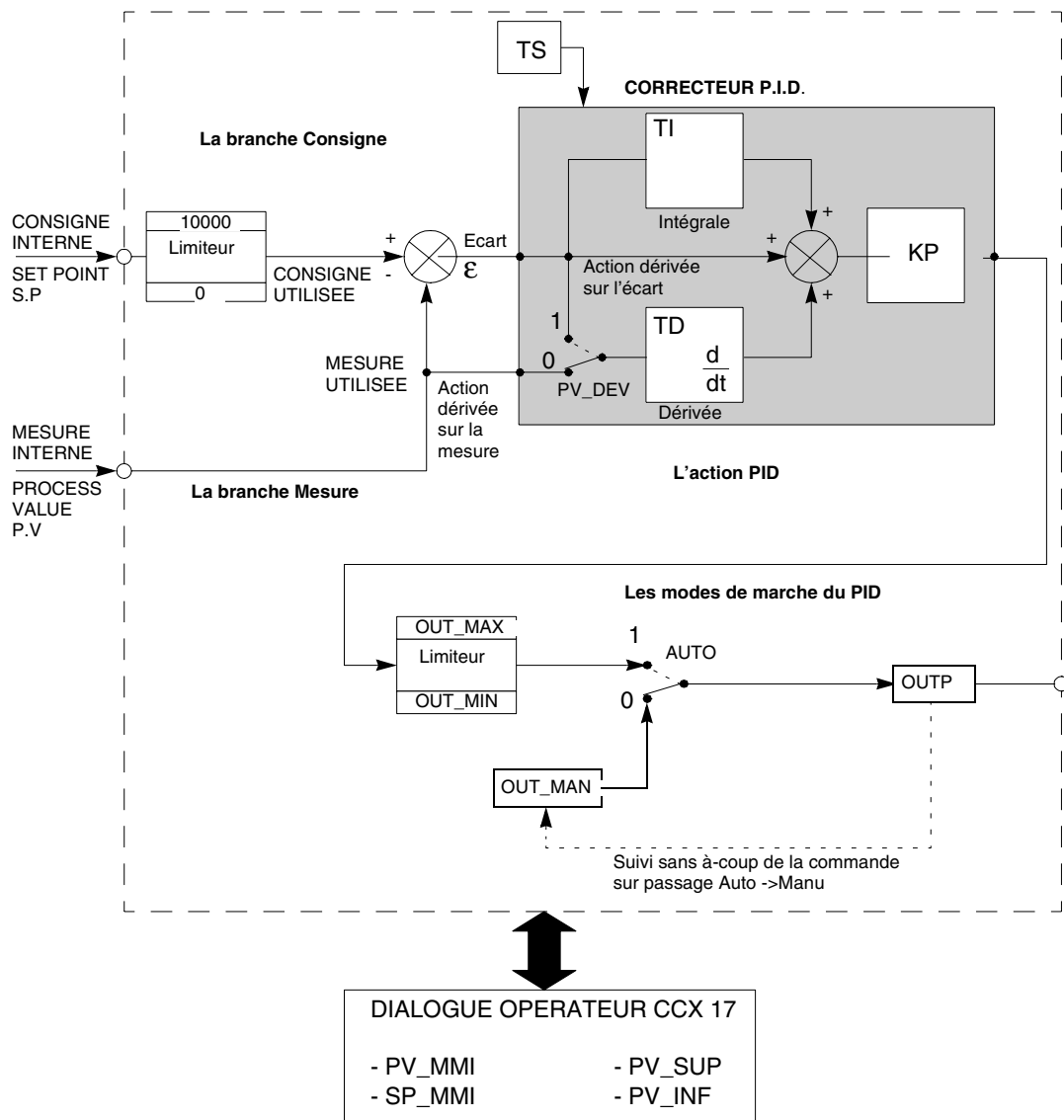
- algorithme PID série / parallèle,
- action direct / inverse (selon le signe du gain KP),
- action dérivée sur mesure ou sur écart,
- limitation haute et basse de la consigne à [0 - 10000],
- limitation haute et basse de la sortie en automatique,
- anti-saturation de l'action intégrale,
- modes de marche Manuel/Automatique sans à coup sur changement,
- contrôle de l'accès PID par le dialogue opérateur,
- fonctionnement en intégrateur pour ($K_P = T_D = 0$).

Note :

- Les paramètres d'affichage utilisés par le CCX 17 sont exprimés en unités physiques,
 - Pour un fonctionnement correct du PID, il est nécessaire de respecter la pleine échelle ; [0-1000] pour la mesure et la consigne.
-

Principe de fonctionnement

Le schéma suivant présente le principe de fonctionnement de la fonction PID.



Note : La description des paramètres utilisés est présentée dans le module (Voir *Programmation de la fonction PID*, p. 353).

Programmation de la fonction PID

Présentation

Les fonctions PID sont des fonctions de base de PL7. A ce titre, elles sont disponibles depuis la bibliothèque de fonctions. Ainsi, est-il possible d'utiliser, depuis les éditeurs langage, l'aide à la saisie d'une fonction PID pour en faciliter sa programmation.

Note : La saisie d'une fonction PID peut se faire dans n'importe quelle tâche périodique (MAST ou FAST). La fonction ne doit pas être conditionnée.

Illustration

L'illustration ci-dessous donne un aperçu de l'écran Fonctions en bibliothèque permettant de mettre en oeuvre la fonction PID.

EF

Informations Fonctions : Paramètres

Famille	V.Bib	V.App	Nom	Commentaire
Fonction Orphée	2.10	-	PID	Régulateur PID mixte
Fonctions temporisation	2.00	-	PID MMI	Gestion du dialogue opérateur dédié sur CCX17 des PID
GRAFCET	1.00	-	PWM	Modulation en largeur d'impulsion d'une grandeur numérique
Réels simple précision	2.22	-	SERVO	Etage de sortie de PID pour commande de vanne TOR
Régulation	2.01	2.01		
Tableaux d'entiers	2.10	-		

Format d'appel

Paramètres de la FONCTION :

Nom	Type	Nature	Commentaire	Zone de saisie
TAG	STRING	IN	Libellé du PID (8 car), utilisé par DOP sur CCX17	"TEMP"
UNIT	STRING	IN	Unité de la mesure (6 car), utilisé par le DOP>>	"DEGRES"
PV	WORD	IN	Mesure, format [0; +10000]	%MW10
OUT	WORD	OUT	Sortie, format [0; +10000]	%MW11

Visualisation de l'appel

PID ("TEMP","DEGRES",%MW10,%MW11,TRIG_PROD_A,%MW20:43)

Syntaxe

La syntaxe d'appel de la fonction PID est :

PID(TAG,UNIT,PV,OUT,AUTO,PARA)

Paramètres de la fonction PID Le tableau ci-dessous présente les différents paramètres de la fonction PID.

Paramètre	Type	Nature IN = Entrée OUT = Sortie	Valeur par défaut	Description
TAG	8 caractères maximum ou %MBi:L avec L inf. ou égal à 8	IN	-	Nom du PID utilisé par le CCX 17.
UNIT	6 caractères maximum] ou %MBi:L avec L inf. ou égal à 6	IN	-	Unité de mesure du PID utilisé par le CCX 17.
PV	%MWi ou %IWxy.i.j	IN	-	Entrée représentant la mesure pour la fonction.
OUT	%MWi ou %QWxy.i.j	OUT	0	Sortie analogique du PID. Si TI = 0, un offset de 5000 est ajouté à la sortie OUT en mode Auto.
AUTO	%Mi , %lxy.i ou %Qxy.i	IN / OUT	0	Mode de marche du PID et du CCX 17. 0 : manuel, 1 = Auto.
PARA	%MWi:43	IN / OUT	-	(Voir tableau ci-dessous pour le détail de la table PARA).

Le tableau ci-dessous présente les différents paramètres de la table PARA :

Paramètre	Rang	Fonction
SP	%MWi	Consigne interne au format 0/10000
OUT_MAN	%MW(i+1)	Valeur de la sortie manuelle du PID (entre 0 et 10000)
KP	%MW(i+2)	Gain proportionnel du PID (x100), signé sans unité (-10000<KP<+10000). Le signe de Kp détermine le sens d'action du PID (négatif : sens direct, positif sens inverse)
TI	%MW(i+3)	Temps d'intégrale du PID (entre 0 et 20000) exprimé en 10 ⁻¹ seconde
TD	%MW(i+4)	Temps de dérivée du PID (entre 0 et 10000) exprimé en 10 ⁻¹ seconde

Paramètre	Rang	Fonction
TS	%MW(i+5)	Période d'échantillonnage du PID (entre 1 et 32000) exprimée en 10^{-2} seconde. La période d'échantillonnage réelle sera le multiple de la période de la tâche dans laquelle est implanté le PID le plus proche de TS
OUT_MAX	%MW(i+6)	Limite supérieure de la sortie du PID en automatique. (entre 0 et 10000)
OUT_MIN	%MW(i+7)	Limite inférieure de la sortie du PID en automatique. (entre 0 et 10000)
PV_DEV	%MW(i+8):X0	Choix action dérivée 0 = sur mesure, 1 = sur écart
NO_BUMP	%MW(i+8):X4	Mode avec ou sans à coups. 0 = avec à coups, 1 = sans à coups
DEVAL_MMI	%MW(i+8):X8	= 1 : inhibe la prise en compte de le PID par le dialogue opérateur. = 0 : le PID est exploité par le dialogue opérateur. Ce bit permet de ne pas faire les conversions d'échelle sur les PID non exploités par le CCX_17,et de sélectionner les PID exploités, surtout dans le cas de plus de 9 PID dans l'application PL7.
PV_SUP (CCX 17)	%MW(i+9)	Borne supérieure de l'étendue de l'échelle de la mesure, en unité physique (x100) (entre -9 999 999 et + 9 999 999).
PV_INF (CCX 17)	%MD(i+11)	Borne inférieure de l'étendue de l'échelle de la mesure, en unité physique (x100) (entre -9 99 999 et + 9 999 999).
PV_MMI (CCX17)	%MD(i+13)	Image de la mesure en unité physique (x100)
SP_MMI (CCX 17)	%MD(i+15)	Consigne opérateur et image de la consigne, en unité physique (x100)

Note :

- Les autres paramètres qui sont utilisés pour la gestion interne du PID ne doivent jamais être modifiés par l'application.
- Les valeurs utilisées par le CCX 17 sont multipliées par 100 afin de permettre un affichage avec 2 chiffres après la virgule sur le CCX 17 (le CCX 17 n'exploite pas le format flottant mais gère un format à virgule fixe).

Règles

Il n'y a pas d'alignement de la consigne interne sur la mesure en mode manuel.

Les mises à l'échelle n'ont lieu que sur modification d'une des consignes (SP ou DOP_SP).

L'algorithme sans action intégrale (TI = 0) effectue l'opération suivante :

Pour	Alors la sortie ...	Avec ...
$\varepsilon t = SP - PV$	$OUT = KP [\varepsilon t + Dt] / 100 + 5000$	Dt= action dérivée

L'algorithme avec action intégrale (TI <0) effectue l'opération suivante ::

Pour	Alors la sortie ...	Avec ...
$\varepsilon t = SP - PV$	$\Delta OUT = KP [\Delta \varepsilon t + (TS/10.TI). \varepsilon t + \Delta Dt] / 100$ $OUT = OUT + \Delta OUT$	Dt= action dérivée

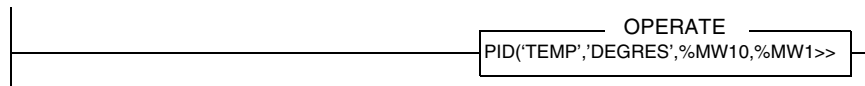
Sur démarrage à froid, le PID repart en manuel, sortie à 0. Pour imposer le mode automatique ou une sortie manuelle non nulle après un démarrage à froid, il faudra programmer la séquence d'initialisation **après** l'appel du PID.

Exemples

Les exemples proposés ci-dessous sont réalisés en langage à contact (Ladder).

Cas où le dialogue opérateur régulation est utilisé (DEVAL_MMI = 0)

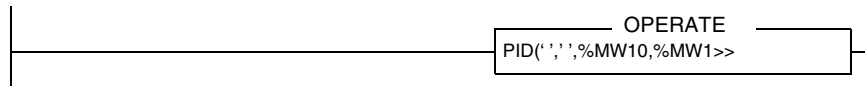
(* Correction PID sur la boucle de régulation de température *)



avec PID('TEMP','DEGRES',%MW10,%MW11,%M10,%MW20:43)

Cas où il n'y a pas de dialogue opérateur DEVAL_MMI = 1.

(* Correction PID sur la boucle de régulation sans DOP intégré



avec PID('','','%IW3.1,%QW4.0,%M10,%MW20:43)

Note : Dans cet exemple, les paramètres TAG et UNIT n'ont pas de sens, il suffit alors de mettre uniquement les côtes.

Fonction PWM

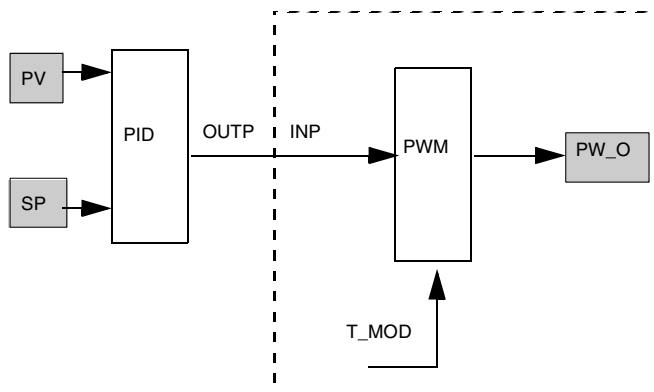
Généralités

La fonction PWM permet de faire de la régulation par largeur d'impulsion sur une sortie TOR. C'est une fonction qui met en forme la sortie du PID.

La largeur des impulsions dépend de la sortie du PID (entrée INP de la fonction PWM) et de la période de modulation.

Principe de fonctionnement

Le synoptique de fonctionnement de la fonction est le suivant :



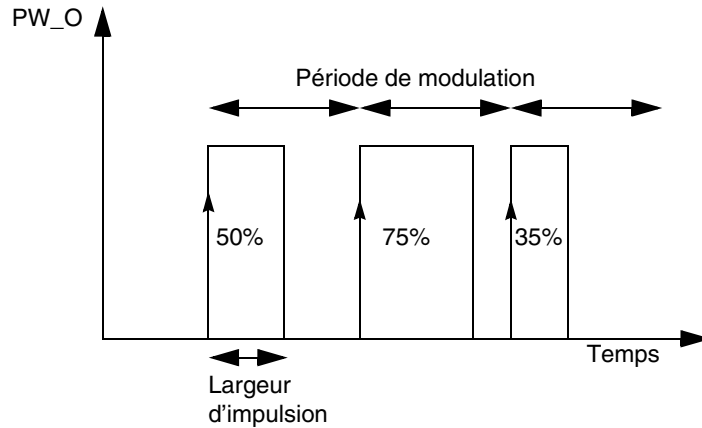
Note : La description des paramètres utilisés est présentée dans le module (Voir *Programmation de la fonction PWM*, p. 359).

Largeur d'impulsions

A chaque TOP de la période de modulation T_{MOD} , la durée d'activation en 10^{-3} seconde de la sortie PW_O est calculée suivant la formule :

Etat 1 du créneau (exprimé en 10^{-2} secondes) = $INP * T_{MOD} / 1000$

Le chronogramme suivant illustre cette formule :

**Règles pratiques**

$T_{MOD} = TS$ (où TS est la période d'échantillonnage du PID amont),
La Période de la tâche courante (exprimée en 10^{-3} seconde) est égale à :
(Résolution désirée) * $10 * T_{MOD}$.

Le PID est dans la tâche MAST, la période de la MAST est de $50 * 10^{-3}$ s,
 $TS = 500 * 10^{-2}$ s et la résolution désirée est de 1/50 (une durée de T_{MOD} doit contenir au moins 50 périodes de la tâche courante).

On prend $T_{MOD} = TS = 500$.

La période de la tâche où est implanté le PWM doit donc être inférieure à $500 * 10 / 50 = 100 * 10^{-3}$ s.

La fonction PWM peut donc être programmée dans la tâche MAST.
la résolution sera de 1/100.

Programmation de la fonction PWM

Introduction

La fonction PWM est une fonction de base de PL7. A ce titre, elle est disponible depuis la bibliothèque de fonctions.
Ainsi, est-il possible d'utiliser, depuis les éditeurs langage, l'aide à la saisie d'une fonction PWM pour en faciliter sa programmation.

Note : La saisie d'une fonction PWM peut se faire dans n'importe quelle tâche périodique (MAST ou FAST). La fonction ne doit pas être conditionnée

Illustration

L'illustration ci-dessous donne un aperçu de l'écran Fonctions en bibliothèque permettant de mettre en oeuvre la fonction PWM.

EF

Informations Fonctions : Paramètres

Famille	V.Bib	V.App	Nom	Commentaire
Fonction Orphée	2.10	-	PID	Régulateur PID mixte
Fonctions temporisation	2.00	-	PID MMI	Gestion du dialogue opérateur dédié sur CCX17 des PID
GRAFCET	1.00	-	PWM	Modulation en largeur d'impulsion d'une grandeur numérique
Réels simple précision	2.22	-	SERVO	Etage de sortie de PID pour commande de vanne TOR
Régulation	2.01	2.01		
Tableaux d'entiers	2.10	-		

Format d'appel

Paramètres de la FONCTION :

Nom	Type	Nature	Commentaire	Zone de saisie
INP	WORD	IN	Grandeur numérique à moduler	%MW11
PW_0	EBOOL	OUT	Sortie TOR rapport cyclique égal à la valeur de INF	%Q6.3
PARA	AR_W	IN/OUT	Paramètres de PWM (table de 5 mots)	%MW90:5

Visualisation de l'appel

PWM (%MW11,%Q6.3,%MW90:5)

Syntaxe

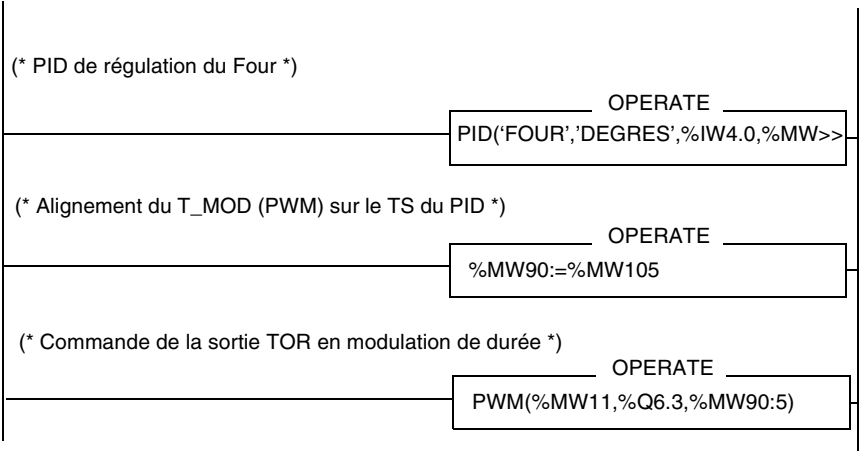
La syntaxe d'appel de la fonction PWM est :

PWM(INP,PW_0,PARA)

Paramètres de la fonction PWM Le tableau ci-dessous présente les différents paramètres de la fonction PWM.

Paramètre	Type	Nature IN = Entrée OUT = Sortie	Description
INP	%MWi	IN	Valeur analogique à moduler en largeur d'impulsion (format [0 - 10000])
PW_0	%Qxy.i ou %Mi	OUT	Sortie logique (TOR) dont le rapport de forme est l'image de l'entrée INP
PARA	%MWi:5	IN / OUT	Période de modulation exprimée en 1/100e de secondes (entre 0 et 32767).T_MOD doit être supérieure ou égale à la période de la tâche courante, et est ajustée par le système pour être un multiple entier de celle-ci. Table de 5 mots dont le premier mot correspond au paramètre T_MOD. Les suivants sont utilisés en interne par la fonction et ne doivent jamais être modifiés par l'application

Exemples L'exemple proposé ci-dessous est réalisé en langage à contact (Ladder).



avec PID('FOUR','DEGRES',%IW4.0,%MW11,%M10,%MW100:43)

Fonction SERVO

Généralités

La fonction SERVO permet de faire de la régulation avec un actionneur de type moteur piloté en 2 actions TOR (UP et DOWN).

Note : Elle doit être obligatoirement connectée en cascade avec la sortie analogique d'un PID. Elle ne peut être utilisée seule.

Lorsqu'une recopie de position existe, un asservissement de la position de la vanne est effectué, à partir des entrées INP (consigne) et POT (mesure de position).

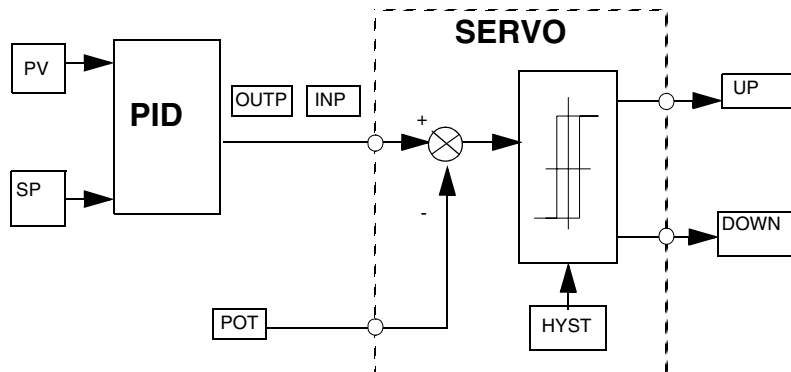
Lorsque la recopie n'existe pas physiquement, l'algorithme n'utilise plus la sortie absolue du PID mais la variation de sortie. La sortie UP (ou DOWN, selon le signe de la variation) est mise à 1 pendant un temps proportionnel au temps d'ouverture de l'actionneur, et à la valeur de la variation. De plus on introduit la notion de temps minimum d'impulsion.

Principe de fonctionnement avec recopie de position

La fonction SERVO effectue un asservissement de la position du moteur en fonction d'une consigne de position INP issue de la sortie d'un PID au format [0 -10000] et d'une mesure de position POT.

L'algorithme d'asservissement est un relais avec hystérésis.

Dans ce cas, les paramètres PID, T_MOTOR et T_MINI ne sont pas utilisés.



Note : La description des paramètres utilisés est présentée dans le module (Voir *Programmation de la fonction SERVO*, p. 364).

Principe de fonctionnement sans recopie de position (POT= - 10000)

Dans ce cas la fonction SERVO se synchronise avec le PID en amont par le biais de la table des paramètres du PID, passée en paramètre à la fonction SERVO.

L'algorithme reçoit en entrée la variation de sortie du PID et la convertit en durée d'impulsion, selon la formule :

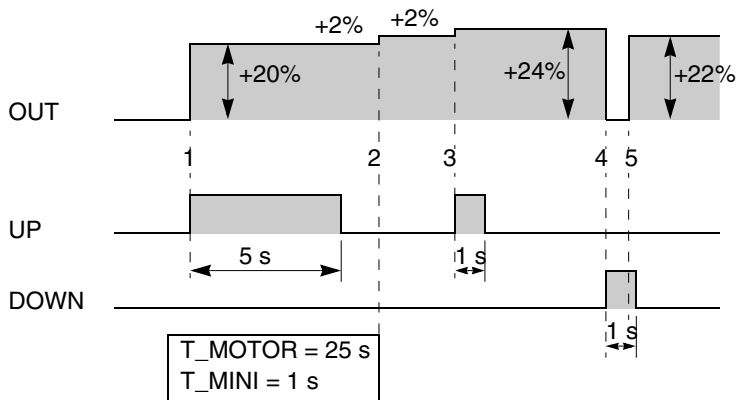
$$T_IMP \text{ (exprimé en } 10^{-3} \text{ s)} = OUT \times T_MOTOR / 1000$$

La durée obtenue s'ajoute à la durée restante des cycles précédents : en effet ce qui n'est pas "consommé" lors d'un cycle est mémorisé pour les cycles suivants. Cela assure un bon fonctionnement notamment sur variation brusque de la commande (ex : échelon de consigne du PID) et en mode manuel.

Note : La description des paramètres utilisés est présentée dans le module (Voir *Programmation de la fonction SERVO*, p. 364).

Exemple

L'exemple proposé ci-dessous est réalisé en langage à contacts (Ladder).



Légende :

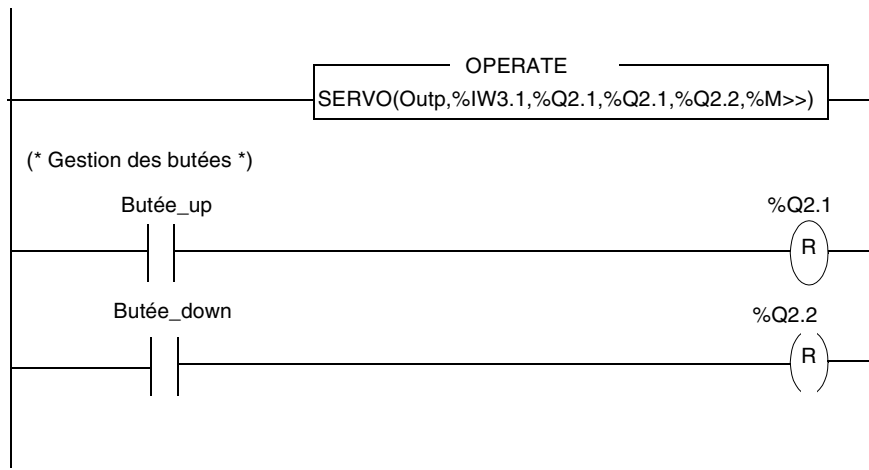
1. La variation de la sortie du PID est de +20% (l'impulsion $T_MOTOR = 25$ s pour une variation de 100%), dans ce cas l'impulsion affecte la sortie UP pour une durée de 5 s,
2. La variation du PID est de +2%, ce qui correspondrait à une impulsion de 0,5 s. Cette impulsion est inférieure à $T_MINI (=1 \text{ s})$, elle n'affecte pas les sorties,
3. Une seconde variation de +2% apparaît, la fonction cumule cette variation avec la précédente (qui correspondait à une variation inférieure à la valeur minimal) pour son calcul, ce qui correspond à une variation positive globale de +4%, et donc à une impulsion de 1 s sur la sortie UP,

4. Une variation de -24% apparaît, l'impulsion lancée est donc de 6 s sur la sortie DOWN,

5. Avant l'écoulement de la seconde suivante, une autre variation de +22% ramène le système à une variation globale de 2% < à la variation de T_MINI (4%). La fonction termine d'effectuer l'impulsion minimale de 1 s.

Note 1 : La fonction SERVO ne gère pas de butées de position, elles doivent être gérées par l'application. En cas de détection de butée, il faut forcer la sortie correspondante à 0 (UP pour la butée haute, DOWN pour la butée basse).

Exemple : (réalisé en langage à contacts (Ladder))



Note 2 : Le passage du mode de fonctionnement avec recopie au mode sans recopie est possible (par ex : sur défaut de recopie, passage au mode sans recopie).

Programmation de la fonction SERVO

Introduction

La fonction SERVO est une fonction de base de PL7. A ce titre, elle est disponible depuis la bibliothèque de fonctions.
Ainsi, est-il possible d'utiliser, depuis les éditeurs langage, l'aide à la saisie d'une fonction SERVO pour en faciliter sa programmation.

Note : La saisie d'une fonction SERVO peut se faire dans n'importe quelle tâche périodique (MAST ou FAST). La fonction ne doit pas être conditionnée.

Illustration

L'illustration ci-dessous donne un aperçu de l'écran Fonctions en bibliothèque permettant de mettre en oeuvre la fonction SERVO.

EF

Informations Fonctions : Paramètres

Famille	V.Bib	V.App	Nom	Commentaire
Réels simple précision	2.22	-	PID	Régulateur PID mixte
Régulation	2.01	-	PID_MMI	Gestion du dialogue opérateur dédié sur CCX17 des PID
Tableaux d'entiers	2.00	-	PWM	Modulation en largeur d'impulsion d'une grandeur numérique
Tableaux de bits	2.00	-	SERVO	Etage de sortie de PID pour commande de vanne TOR
Tableaux de réels	2.10	-		
Tableaux d'entiers doubles	2.00	-		

Format d'appel

Paramètres de la PROCEDURE :

Nom	Type	Nature	Commentaire	Zone de saisie
INP	WORD	IN	Consigne de position, format [0;10000] (à conn>>	OUTP
POT	WORD	IN	Recopie de position, format [0;10000] [-10000>>	-10000
UP	EBOOL	OUT	Sortie TOR, sens de marche UP	%Q2.1
DOWN	EBOOL	OUT	Sortie TOR, sens de marche DOWN	%MW100:43

Visualisation de l'appel

SERVO (OUTP;-10000;%Q2.1;%MW100:43;%MW180:10)

Syntaxe

La syntaxe d'appel de la fonction SERVO est :

SERVO(INP,POT,UP,DOWN,PID,PARA)

Paramètres de la fonction SERVO Le tableau ci-dessous présente les différents paramètres de la fonction SERVO.

Paramètre	Type	Nature IN = Entrée OUT = Sortie	Description
INP	%MWi	IN	Consigne de position (format [0 - 10000]) à connecter obligatoirement à la sortie du PID.
POT	%MWi ou direct	IN	Recopie de position (format [0 - 10000]) 0 : vanne fermée; 10000 : vanne ouverte. Si la recopie n'existe pas. POT doit être initialisé à -10000. Cette valeur particulière signifie "pas de recopie".
UP	%Qxy.i ou %Mi	OUT	Signal de sortie pour le sens de marche UP du moteur.
DOWN	bit de type %Q ou %M	OUT	Signal de sortie pour le sens de marche DOWN du moteur.
PID	%MWi:43	IN / OUT	Table du paramètre PARA du PID amont. Utilisé s'il n'y a pas de mots de recopie pour la synchronisation avec le PID amont. Voir <i>Paramètres de la fonction PID</i> , p. 354.
PARA	%MWi:10	IN / OUT	(Voir tableau ci-dessous pour le détail de la table PARA).

Le tableau ci-dessous présente les différents paramètres de la table PARA :

Paramètre	Rang	Fonction
T_MOTOR	%MWi	Temps d'ouverture vanne exprimé en 10^{-2} s. Utilisé si la recopie n'existe pas (POT = -10000).
T_MINI	%MW(i+1)	Temps minimal d'impulsion exprimé en 10^{-2} s. Utilisé si la recopie n'existe pas (POT = -10000).
HYST	%MW(i+2)	Valeur de l'hystérésis au format [0 - 10000]. Utilisé si la recopie n'existe (POT : [0 - 10000]).

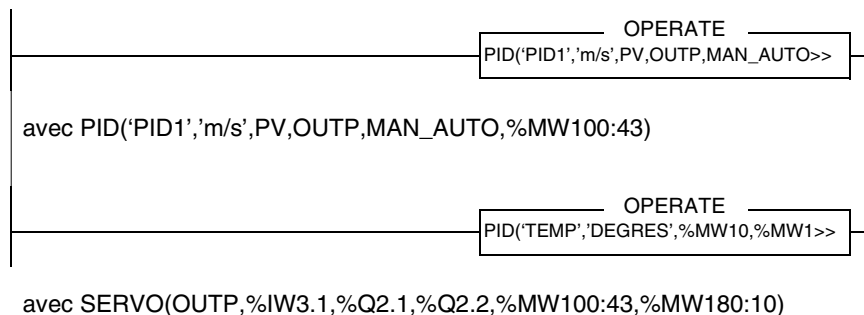
Note :

- Les autres paramètres qui sont utilisés pour la gestion interne de la fonction ne doivent jamais être modifiés par l'application.
- Tous les paramètres sont obligatoires, indépendamment du mode de fonctionnement.

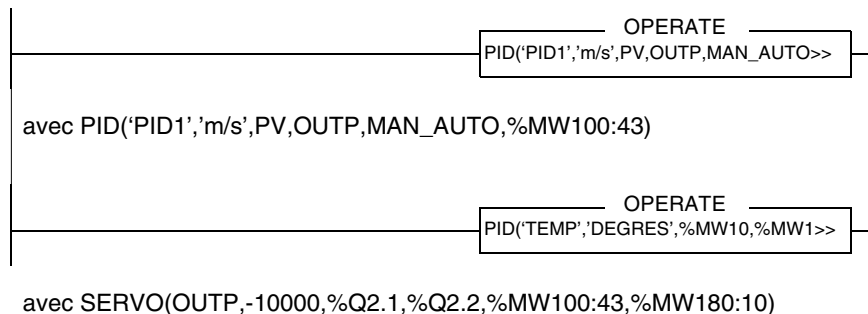
Exemples

Les exemples proposés ci-dessous sont réalisés en langage à contact (Ladder).

Cas avec recopie de position.



Cas sans recopie de position.



Comportement des fonctions dans les modes de marche

Introduction

Ce paragraphe décrit le comportement des fonctions dans les différents cas de démarrage :

- démarrage à froid (nouvelle application, changement de cartouche...),
- reprise à chaud (retour secteur, sans changement de contexte application),
- première exécution après ajout d'une fonction par modification en connecté.

Démarrage à froid

Ce type de démarrage intervient pour une nouvelle application, un changement de cartouche

Sur démarrage à froid, l'automate peut démarrer automatiquement en RUN (selon la configuration de l'application). Les fonctions correcteurs ont un comportement sécurité: mode manuel, sorties à 0. De plus cela permet de passer l'automate en RUN sans effectuer de réglage du PID, puis de faire sa mise au point avec le CCX 17 (le réglage ne peut se faire qu'en RUN).

Reprise à chaud

Ce type de reprise intervient pour un retour secteur, sans changement de contexte application.

Sur retour secteur après une coupure (indépendamment de sa durée) et si le contexte application n'est pas perdu ou modifié, les fonctions repartent dans l'état avant coupure. Si l'utilisateur souhaite un autre comportement, il est de sa responsabilité de tester le bit système %S1 et d'y associer le traitement voulu (forçage en mode manuel...).

Note : L'horodateur de l'automate permet de connaître la durée de la dernière coupure.

Ajout en connecté d'un nouvel appel

Suite à l'ajout d'un nouvel appel de fonction de régulation en connecté, une initialisation identique au cas de la reprise à froid est effectuée.

Note : Pour être vue comme une nouvelle fonction, celle-ci doit utiliser une nouvelle table de paramètres. Donc le retrait d'un PID, suivi de l'ajout d'un PID utilisant la même table de paramètres n'est pas considéré comme un ajout de nouveau PID. Dans ce cas le PID s'exécute dans l'état et avec les paramètres du PID précédent.

