

République Algérienne Démocratique et Populaire
Ministère de L'Enseignement Supérieur et de la Recherche Scientifique

UNIVERSITE MOULOUD MAMMARI DE TIZI-OUZOU



FACULTE DE GENIE ELECTRIQUE ET D'INFORMATIQUE
DEPARTEMENT D'AUTOMATIQUE

**Mémoire de Fin d'Etude
de MASTER ACADEMIQUE**
Spécialité : **Commande des systèmes**
Filière : **Génie Electrique**

Présenté par
Samir AIT ABDELLAH

Thème
**Implémentation de la commande RST
sous un API S7-300 pour le contrôle de
la station de température PUT-1
d'Electronica Veneta**

Mémoire soutenu publiquement le mercredi 30/09/2015 à 8h30 devant le jury composé de

M Prénom NOM

MAB, Belkacem MOULA, UMMTO, Président

M Prénom NOM

MAA, Rezki HADDOUCHE UMMTO, Rapporteur

M Prénom NOM

MAA, Mourad ALLAD UMMTO, Examineur

M Prénom NOM

MAA, Moussa CHARIF UMMTO, Examineur

REMERCIEMENT

Je tiens à remercier Dieu le tout puissant de m'avoir donné le courage, la force et la volonté pour réaliser ce travail.

Un grand merci à **Mme KHOUAS** la responsable de laboratoire d'automates programmables et de la régulation industrielle qui nous a permis de travailler au sein de ce laboratoire

Mes chaleureux remerciements à mon promoteur **Mr R. HADDOUCHE** pour sa confiance, sa disponibilité, sa patience et son aide précieuse tout au long de mon projet.

Mes vives reconnaissances vont également à tous les membres de jury d'avoir acceptés d'examiner mon travail.

A l'ensemble des enseignants de département Automatique.

Sans oublier ceux qui ont participé de près ou de loin à la réalisation de ce travail.

Samir

Dédicaces

Je dédie ce modeste travail à :

Mes très chers parents, ma mère Djouher mon père Said

Mes très adorables frères.

Mes très chères sœurs.

Ma nièce.

« Kamilia »

Toute ma famille.

Tous mes ami(e)s qui ont contribué à la réalisation de ce modeste travail.

Tous mes collègues de la promotion Automatique 2015.

Sommaire

Sommaire

Introduction générale.....	1
Chapitre I Elaboration du Modèle dynamique d'un four électrique	
I.1 Introduction.....	3
I.2 Présentation du four électrique	4
I.2.1 échauffement d'un corps	4
I.2.2 Contrôle du processus de température	6
I.3 Identification du procédé	6
I.3.1 Détermination du modèle continu du procédé	7
I.3.1.1 La caractéristique statique du four	7
I.3.1.2 La caractéristique dynamique du four	9
I.3 Modélisation du four	16
I.4 Etude comparative des modèles obtenu.....	19
I.5 Conclusion	20
Chapitre II Application de la Commande RST pour la régulation de température	
II.1 Introduction	21
II.2 Présentation de la commande RST.....	22
II.2.1 description des éléments de la boucle de régulation.....	22
II.2.2 avantages de régulateur RST	22
II.3 Méthode de synthèse d'un régulateur RST	24
II.3.1 Principe.....	24
II.3.2 Différents types de système.....	24
II.3.2.1 processus de type P1.....	24
II.3.2.2 processus de type P2.....	24
II.3.2.3 processus de type P3.....	25
II.3.3 Synthèse de R, S et T	26
II.4 Application de la commande RST pour la régulation de température	28
II.5 Conclusion.....	31

Sommaire

Chapitre III Méthodologie d'implémentation de la commande RST sur un API S7-300

III.1 Introduction	32
III.2 Présentation de l'automate programmable industrielle S7-300.....	32
III.2.1 Domaine d'application.....	33
III.2.2 Structure interne d'un Automate programmable	34
III.2.3 Caractéristiques de l'API S7-300	34
III.2.4 Différents parties de l'API S7-300.....	35
III.2.4.1 Module d'alimentation.....	35
III.2.4.2 Unité centrale	35
III.2.4.3 Les module d'entrée/sortie	35
III.2.4.4 Les coupleurs	36
III.2.4.5 Modules de fonction (FM)	36
III.2.4.6 Fonctionnement de l'API.....	36
III.3 Méthodologie d'implémentation sur API S7-300	36
III.4 Implémentation du régulateur RST pour la régulation de la température	39
III.4.1 Création d'un modèle Simulink et génération d'une source SCL et une DLL Windows	39
III.4.2 Implémentation de la source SCL et création de block « RST_Control » dans Step7.....	43
III.4.2.1 Intégration de la Bibliothèque WinAC ODK dans Step7	43
III.4.2.2 Création de block fonctionnel à partir de la source SCL	44
III.5 Conclusion.....	46
Conclusion générale	47

Introduction générale

Introduction générale

Un système est un assemblage, un ensemble, ou une collection des choses (ou de composants physiques) reliées entre elles de manière à former une entité. Un système de contrôle est un assemblage de composants physiques reliés entre eux de manière à contrôler, diriger, ou réguler lui-même ou d'autres systèmes.

En général et en un sens abstrait, ces systèmes peuvent être n'importe quelles composantes physiques qui peuvent altérer ses environnements d'une certaine manière. En Ingénierie nous considérons uniquement les systèmes qui ont pour une fonction majeure de contrôler, diriger ou réguler.

Nous implantons un asservissement dans un système dès lors que les performances attendues sont trop exigeantes pour que le processus seul les atteigne naturellement, ou lorsque le processus est soumis à des perturbations extérieures qui modifient la grandeur.

Les performances attendues sont principalement la stabilité, la précision et la rapidité de répondre.

La plupart des boucles d'asservissement mettent en œuvre le classique correcteur PID. Bien que relativement facile à régler, le PID ne permet pas toujours d'obtenir les performances requises, en particulier sur les systèmes :

- à retard pur - dont les caractéristiques dynamiques varient au cours du fonctionnement
- d'ordre supérieur à 2
- possédant des zones de non linéarités

Le correcteur numérique RST (dont le PID est un cas particulier) est bien adapté à ces types de systèmes. Il permet :

- de gérer le retard pur.
- de gérer de manières distinctes les dynamiques de suivi de consigne et de rejection de perturbation.

La consigne peut être atteinte selon une trajectoire du 2nd ordre entièrement paramétrable (dépassement, temps de montée)

- d'obtenir un excellent réglage, pour s'affranchir des variations dans la dynamique du système (gain, retard, inertie) Il s'intègre sur toute cible programmable (carte contrôleur, automate programmable, PC ...).

L'Automate Programmable Industriel (API) est un appareil électronique programmable, adapté à l'environnement industriel, qui réalise des fonctions d'automatisme pour assurer la commande de prés actionneurs et d'actionneur à partir d'informations logiques, analogiques ou numérique.

Les API comportent mieux d'avantage par rapport à d'autre support d'implémentation dans l'industrie ; ils sont résistant aux influences externes (la poussière, l'humidité, les vibrations et les parasites électromagnétiques) il permet une intervention rapides du personnel qualifié pour le dépannage ou/et la modification.

Notre travail consiste à identifier un système (four électrique), ensuite la mise en œuvre de la commande RST, puis l'implémenter sous l'API Siemens S7-300.

Chapitre I

Elaboration du Modèle Dynamique d'un four Électrique

Elaboration du modèle dynamique d'un four électrique

I.1 Introduction

Le four électrique à résistances est sans doute le plus connu des appareils électrothermiques. Son utilisation industrielle remonte en effet au début des années 1920 et sa technique n'a cessé de se perfectionner depuis. [1]

Le principe de ce type de four est extrêmement simple : il est constitué d'une enceinte chauffée à l'aide de résistances électriques. La charge à chauffer est placée dans cette enceinte.

Le four à résistances est un équipement à chauffage indirect, la chaleur produite par effet Joule par les résistances est transmise à la charge par rayonnement et convection.

La robustesse et la fiabilité de ces équipements, leur facilité de mise en œuvre, la simplicité de maintenance et leur efficacité énergétique élevée font que les applications industrielles sont fort nombreuses, il n'est guère de secteur industriel qui n'y fasse appel.

L'automaticien a besoin d'un modèle pour concevoir un régulateur à mettre en œuvre afin d'atteindre les objectifs décrits dans un cahier des charges.

Les modèles les plus utilisés en automatique sont les modèles des connaissances et les modèles de représentation. Ainsi, le modèle de connaissance est peu utilisé car les équations physique régissant les processus ne sont pas toujours facile à obtenir, contrairement au modèle de représentation qu'est basé sur la connaissance expérimentale des entrées/sorties, pour élaborer ce modèle on a besoin d'utiliser les méthodes d'indentification. [2]

➤ Modélisation

La modélisation permet de représenter un procédé de façon simplifiée. Cela aide à faire l'analyse. La modélisation implique de faire des hypothèses sur le procédé ou certains de ses paramètres pour pouvoir faire certaines simplifications. Il faut toutefois s'assurer de ne pas négliger des paramètres importants du procédé. [2]

Le développement d'un modèle pour un système physique peut être réalisé pour différentes raisons :

- Avoir une meilleure compréhension des phénomènes
- Dimensionnement d'une installation
- Formation des opérateurs
- Conception du système de commande, cette dernière a pour objectif
 - La mise au point de la stratégie de commande
 - Conception de la loi de commande et son réglage
 - Conception de capteurs logiciels ou estimateur d'état du système.

➤ Identification

L'objectif de l'identification est de calculer les paramètres d'un modèle du procédé, à partir des données expérimentales, de façon à ce que le comportement du procédé et celui de modèle soient identiques, et ceci pour toutes les séquences de variables d'entrées habituellement utilisées. [2]

Dans le cadre du contrôle d'un système industriel, une grande importance est accordée à l'identification du processus. En effet, pour procéder à la commande d'un système réel, il est nécessaire de disposer d'un modèle mathématique qui le caractérise avec une bonne approximation mais qui doit être suffisamment simple pour faciliter son étude.

I.2 Présentation du four électrique

I.2.1 échauffement d'un corps

On décrit par la suite le processus physique de réchauffement d'un corps. un conducteur de chaleur ayant une masse "m" (Kg) et une chaleur spécifique "c" Joule par kilogramme-kelvin ($J \cdot kg^{-1} \cdot K^{-1}$). Si on fournit une quantité de chaleur "Q"(Joule), sa température augmente d'une quantité δT , qui vaut :

$$\delta T = \frac{Q}{m \cdot c} \quad (\text{Kelvin}) \quad (I.1)$$

Dans le système international, on exprime la quantité de chaleur "Q" en Joules et on la calcule en multipliant la valeur de la puissance (supposée constante, exprimé en Watts) donnée à l'élément réchauffant par le temps (secondes) d'application.

Le processus peut avoir lieu dans un creuset, un réservoir de réactifs chimique, un four, etc.

L'unité de température PUT-1 d'Elettronica Veneta présente dans l'équipement a pour centre processus thermique une plaque d'aluminium. Cette dernière permet d'obtenir des températures élevées (jusqu'à 250°C) avec une puissance de chauffage relativement limitée (maximum 100 W) ; en outre elle permet d'amener le processus au régime statique avec une extrême rapidité. L'unité de température PUT-1 comprend les actionneurs thermiques suivants : une résistance électrique à deux éléments (2*50 W, 25 Vca) et un ventilateur (170m³/h , 24 Vcc, 4.5 W). Le ventilateur permet de refroidir rapidement la plaque d'aluminium, afin de pouvoir réaliser les expériences suivantes après peu de temps.

La plaque d'aluminium présente des points de mesure pour les trois types de transducteurs de température industriels fournis avec l'unité de mesure de température (PTC, thermorésistante et thermocouple).

L'unité de température est équipée d'un thermocouple pour réaliser les expériences avec l'API (ou PLC). Il possède en outre un thermomètre au mercure, que l'on emploiera pour donner la température de référence. On insérera ce thermomètre en un point de mesure spéciale se trouvant sur la plaque d'aluminium. L'unité de température est montrée à la figure I.1.

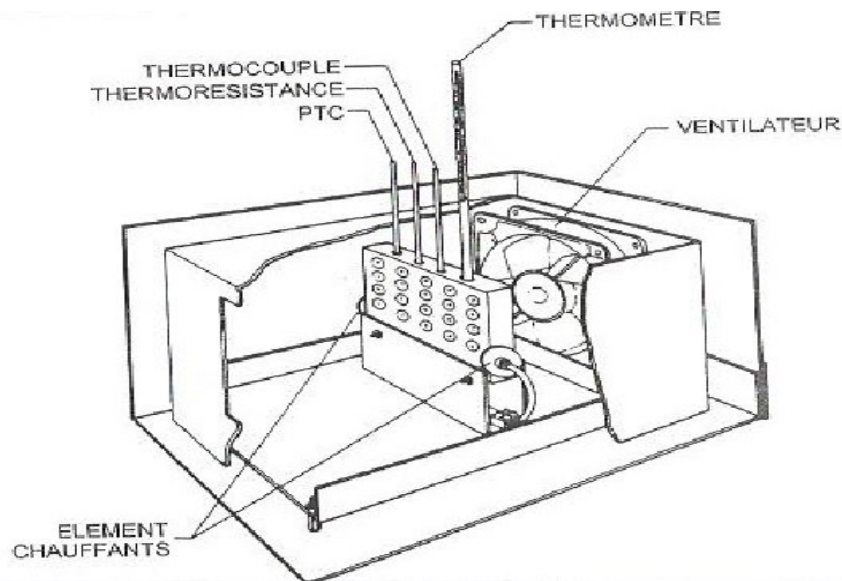


Figure I.1 : schéma des différentes composantes du four

L'unité de température peut fonctionner sur la plage de température entre 0°C à 250 °C

I.2.2 Contrôle du processus de température

Le contrôle automatique de la température est très répandu tant dans le domaine industriel que dans le domaine domestique.

Le schéma fonctionnel de la figure I.2 montre un système typique de contrôle du processus de température.

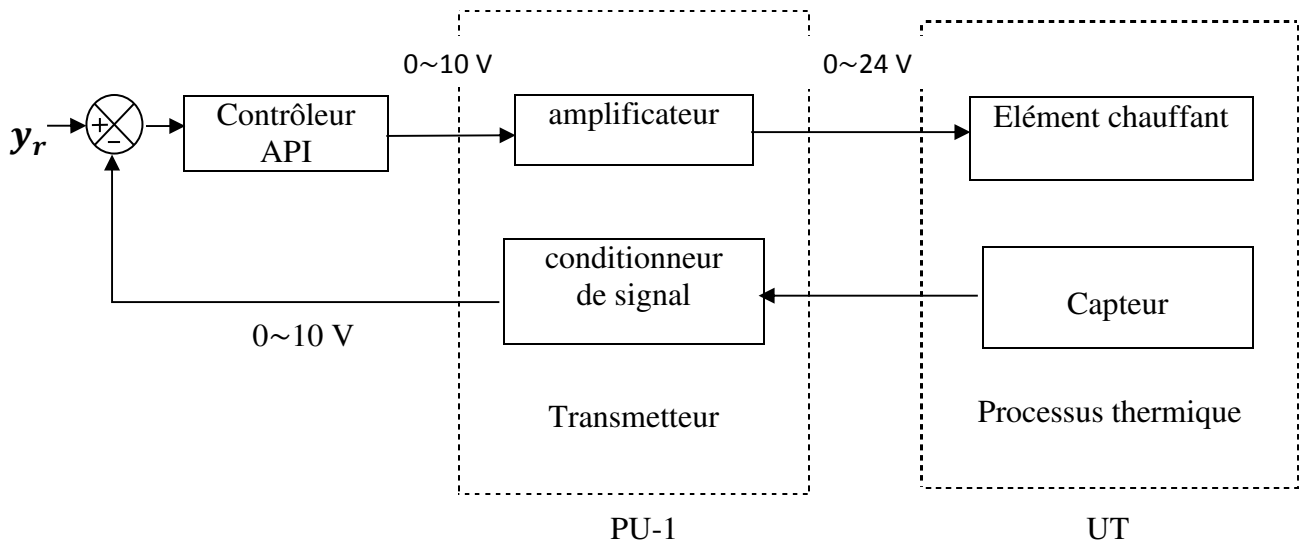


Figure I.2 : schéma fonctionnel de contrôle du four

I.3 Identification du procédé

Une identification, du procédé à commander est un préalable indispensable. Un minimum d'information sur ce dernier est nécessaire pour faire le choix d'une stratégie de commande qui convienne et de régulateur appropriée. [2]

Dans notre travail, pour la mise en œuvre d'une commande par un automate programmable industriel, l'identification est une étape fondamentale, il est important de connaître le modèle précise du procédé pour synthétiser un régulateur.

I.3.1 Détermination du modèle continu du procédé

I.3.1.1 La caractéristique statique du four

Pour relever la caractéristique statique, on applique au four des tensions différentes tout en restant dans la zone de fonctionnement (0 à 10 volts), pour chaque valeur, il faut atteindre le régime permanent pour noter la valeur de la sortie V_m (V) fournis par le thermocouple et la température à l'enceinte du four T_m mesurée à l'aide d'un thermomètre.

Commande U (v)	1.00	2.00	3.00	4.00	5.00	6.00	7.00	8.01
Mesures								
v_m (v)	1.34	3.79	4.95	5.97	7.33	8.97	9.1	9.39
T_m (°C)	34	90	120	152	170	200	223	235

Tableau I.1 : tableau de la caractéristique statique du four

Pour faire apparaître la zone de fonctionnement linéaire du four nous avons besoin de tracer le comportement statique

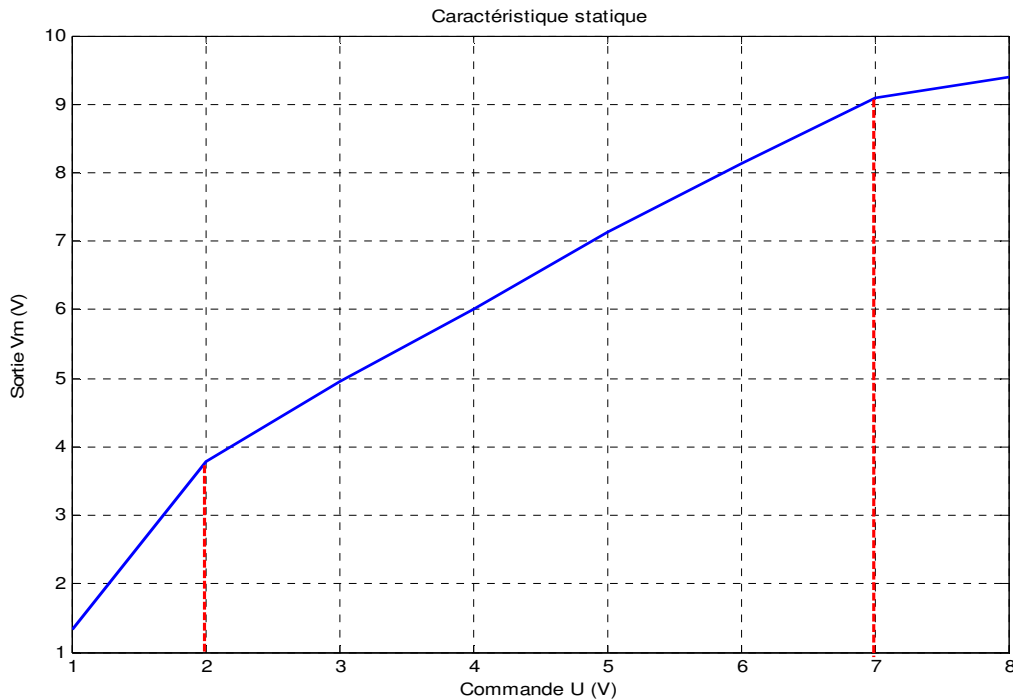


Figure I.2 : Caractéristique statique

Pour faire apparaître le comportement statique la figure I.3 nous donne la température en fonction de différentes commandes.

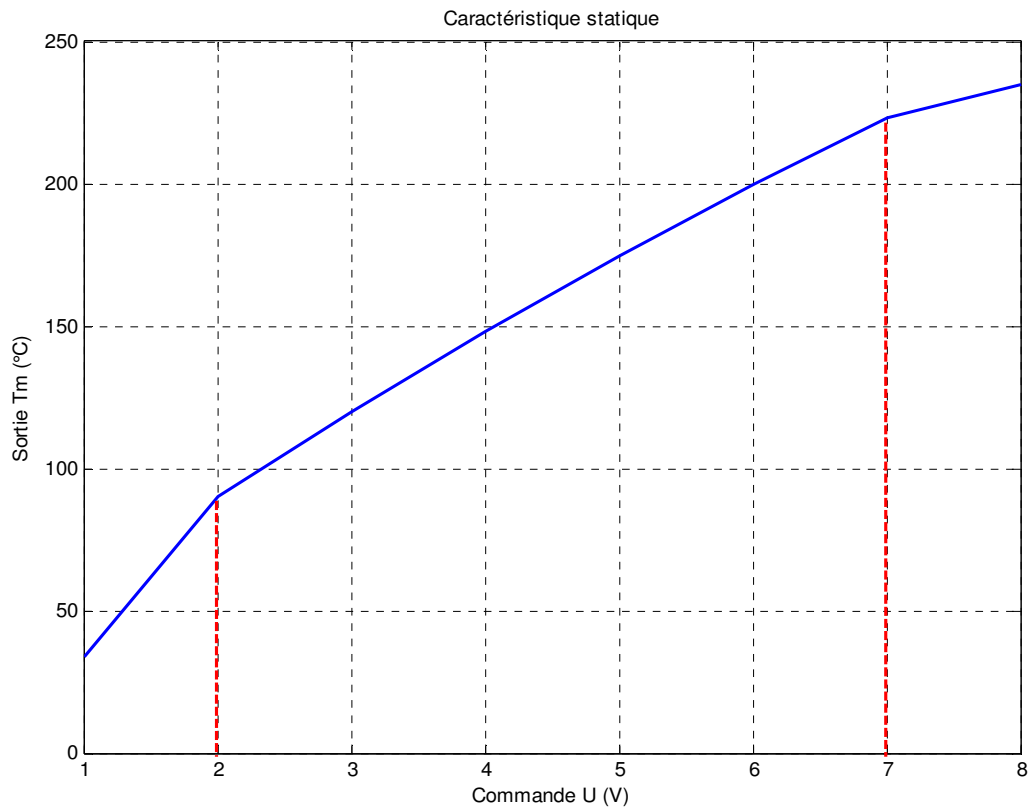


Figure I.3 : caractéristique statique

Remarque : La zone de fonctionnement linéaire est entre (2~7 volts).

I.3.1.2 La caractéristique dynamique du four

Dans ce qui suit, nous allons établir un modèle mathématique du processus thermique étudié en utilisant la méthode d'identification empirique de Broïda. Elle se base sur l'allure de la réponse indicielle du système en boucle ouverte. En effet, dans le cas où cette réponse indicielle ressemble à l'allure de la courbe de figure I.4, nous appliquons la méthode Broïda.

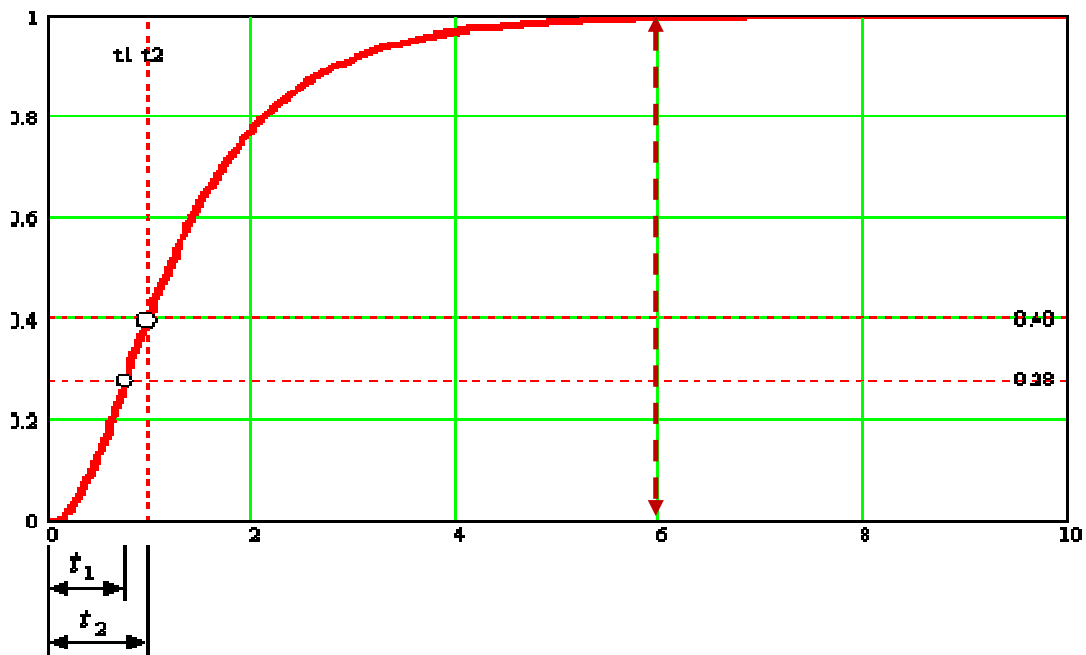


Figure I.4 réponse indicielle en boucle ouverte

Le modèle proposé pour approcher le comportement du système est un premier ordre avec un retard pur. Sa fonction de transfert est donnée par l'équation (I.2)

$$G(P) = \frac{K.e^{-\tau P}}{1+TP} \quad (I.2)$$

Le problème d'identification consistera donc à déterminer les paramètres suivants

T Constante du temps (seconde)

τ temps de retard pur (seconde)

Afin de déterminer des valeurs de ces paramètres, Broïda fait correspondre la réponse indicielle à identifier et la fonction de transfert du 1^{er} ordre affectée d'un retard en deux points t_1 et t_2 d'ordonnées correspondant à 28% et 40% de la valeur finale de la sortie du système.

$$\frac{s(t)}{K.E} = 0.28 \rightarrow \frac{t_1 - \tau}{T} = 0.328 \tag{I.3}$$

$$\frac{s(t)}{K.E} = 0.40 \rightarrow \frac{t_2 - \tau}{T} = 0.510 \tag{I.4}$$

La résolution de ces équations donne

$$T = 5.5(t_2 - t_1) \tag{I.5}$$

$$\tau = 2.8t_1 - 1.8t_2 \tag{I.6}$$

Le gain K est le rapport entre la valeur finale de la sortie et l'entrée appliqué.

Le travail que nous avons fait au laboratoire consiste à faire l'identification du four en appliquant la consigne tout en restant dans la zone de fonctionnement linéaire (de 2 à 7 volts), puis relever les mesures durant le régime transitoire

1^{er} essai 4.00 volt

Résultats de simulation

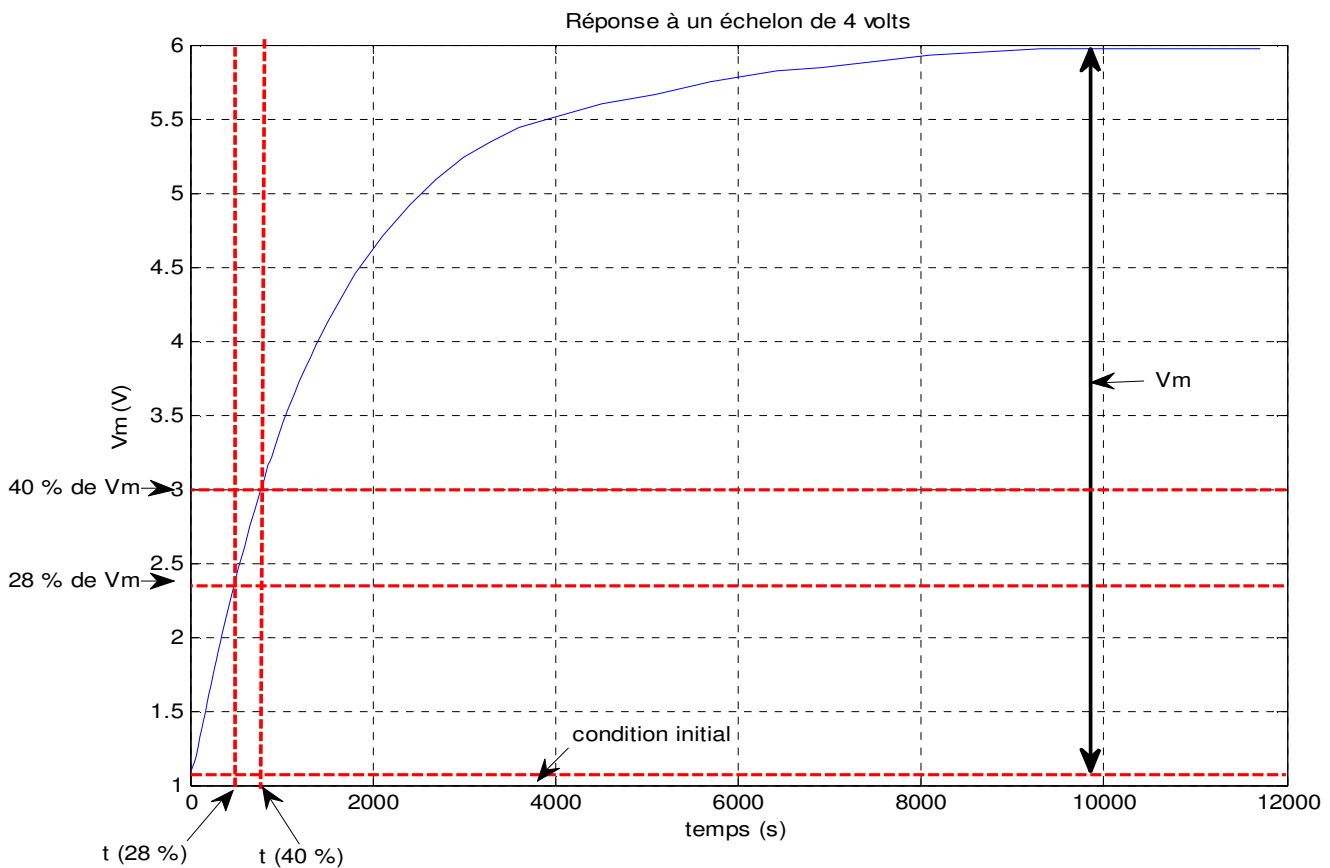


Figure I.4 : réponse à un échelon de 4 volts

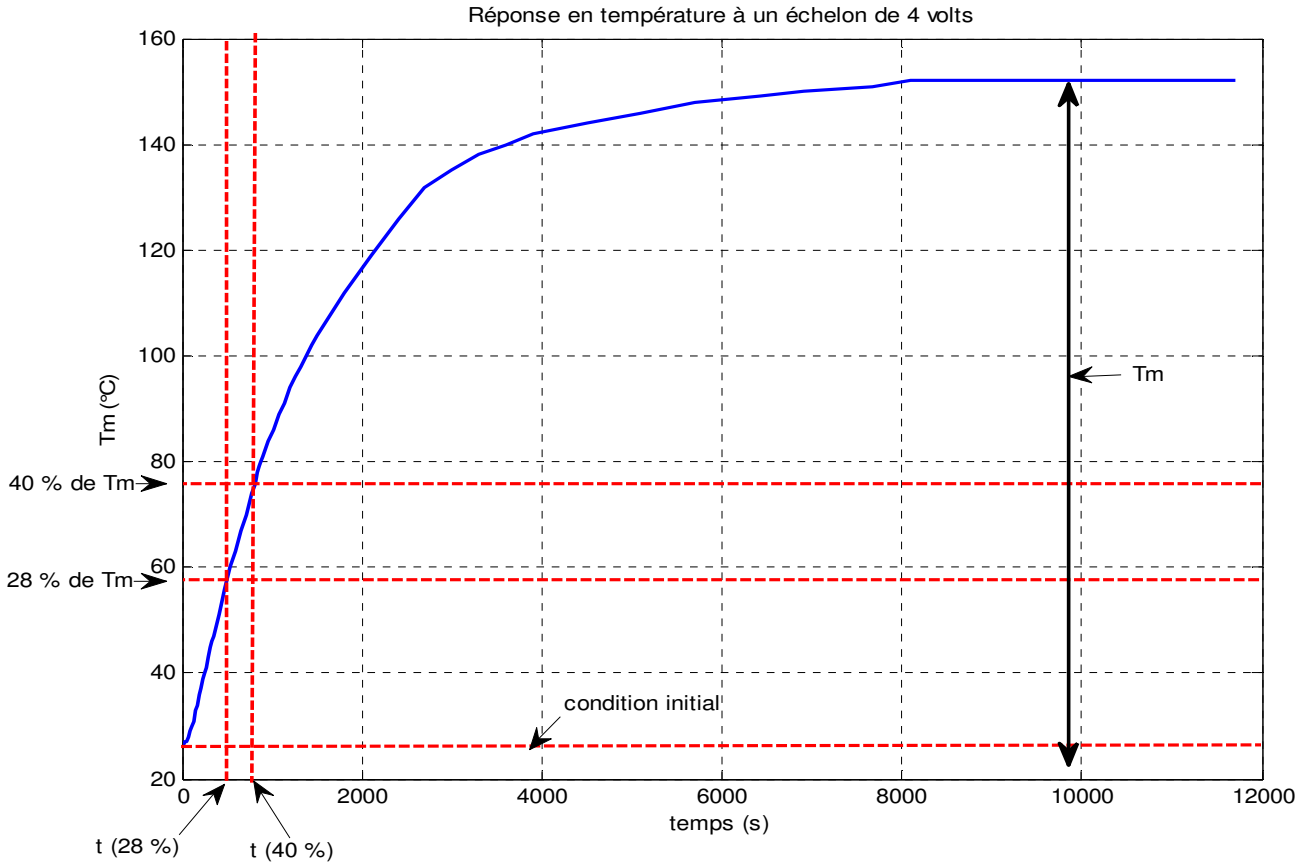


Figure I.5 : réponse en température à un échelon de 4 volts

Application de la méthode de Broïda :

- Détermination du gain K :

En régime permanent ($t \rightarrow \infty$) on a : $E \cdot K = s(t) - s(0) = \Delta s \rightarrow K = \frac{s(t) - s(0)}{E}$ (I.7)

$K=1.22$

- Détermination de T et τ :

On doit d'abord trouver $t_{28\%}$ et $t_{40\%}$:

$$s(t_{28\%}) = \Delta s * 0.28 + s(0) = 4.88 * 0.28 + 1.09 = 2.4564$$

(I.8)

$$s(t_{40\%}) = \Delta s * 0.4 + s(0) = 4.88 * 0.4 + 1.09 = 3.0420$$

(I.9)

A partir de la figure 5 on trouve

$t_{28\%} = 540s$

$t_{40\%} = 800s$

$T=1430s$

$$\tau = 2.8 * 540 - 1.8 * 800$$

$$\tau = 72 \text{ s}$$

$$G(P) = \frac{1.22.e^{-72P}}{1+1430P} \quad (\text{I.10})$$

Nous pouvons approximer ce modèle au modèle de l'équation (I.11), en utilisant la série de Padé.

$$G(P) = \frac{1.22}{(1+1430P)(1+72P)} \quad (\text{I.11})$$

La réponse indicielle de $G(p)$ est illustrée sur la figure I.6

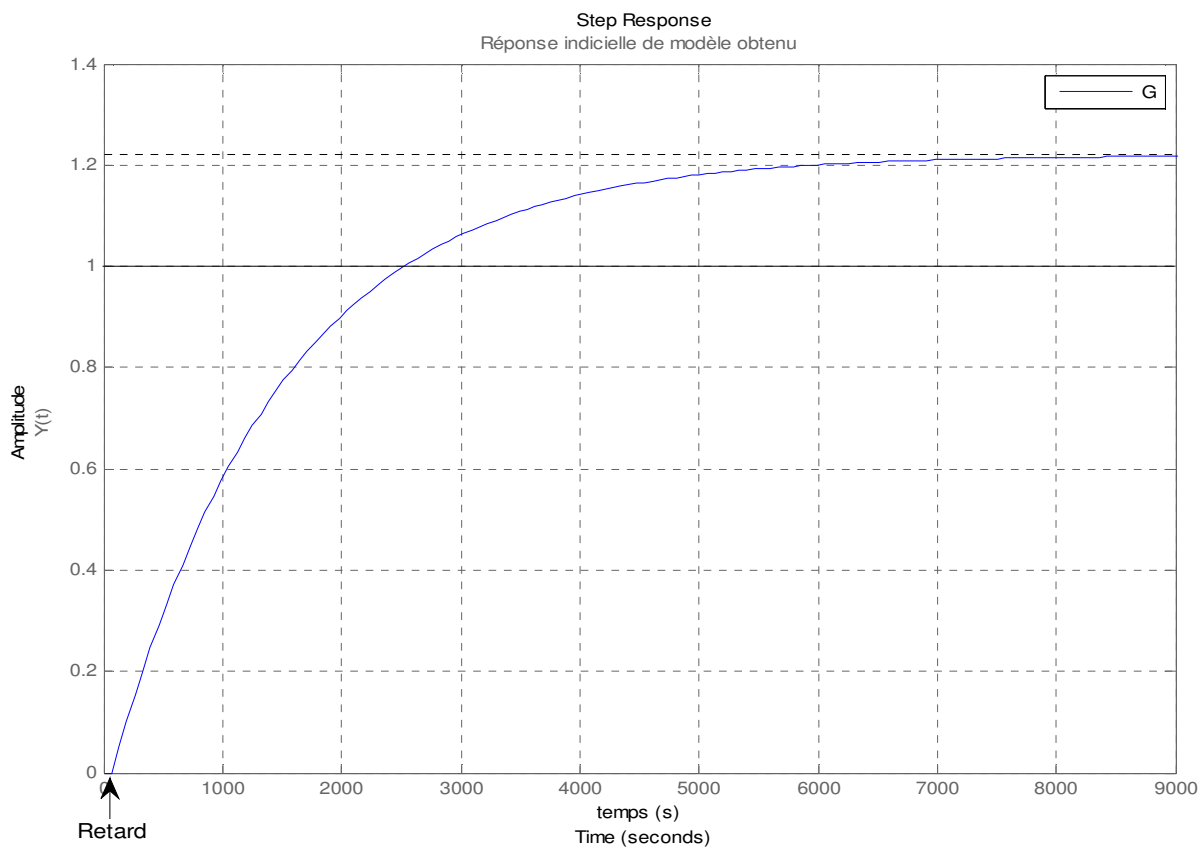


Figure I.6 : réponse indicielle de modèle obtenu

On remarque que la réponse démarre de 0 volts car la condition initiale (la température ambiante) n'est pas prise en compte, on peut l'ajouter au modèle obtenu pour pouvoir comparer la réponse réelle du four à celle du modèle obtenu.

2^{ème} Essai consigne de 6 volts

L'application d'une tension de 6 volt a permis d'enregistrer la réponse du four présenté sur la figure I.6 et I.7

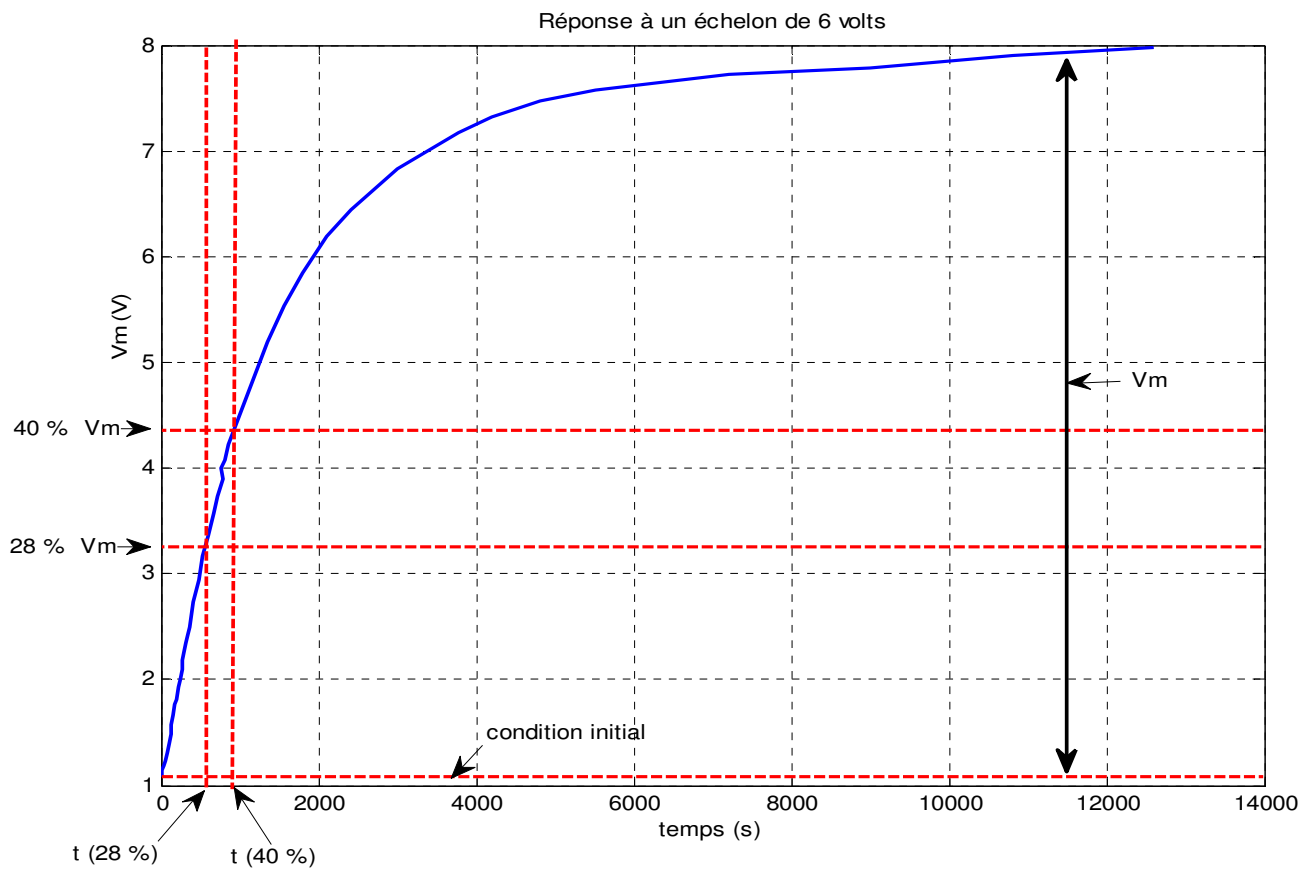


Figure I.6 : réponse du four à un échelon de 6 volts

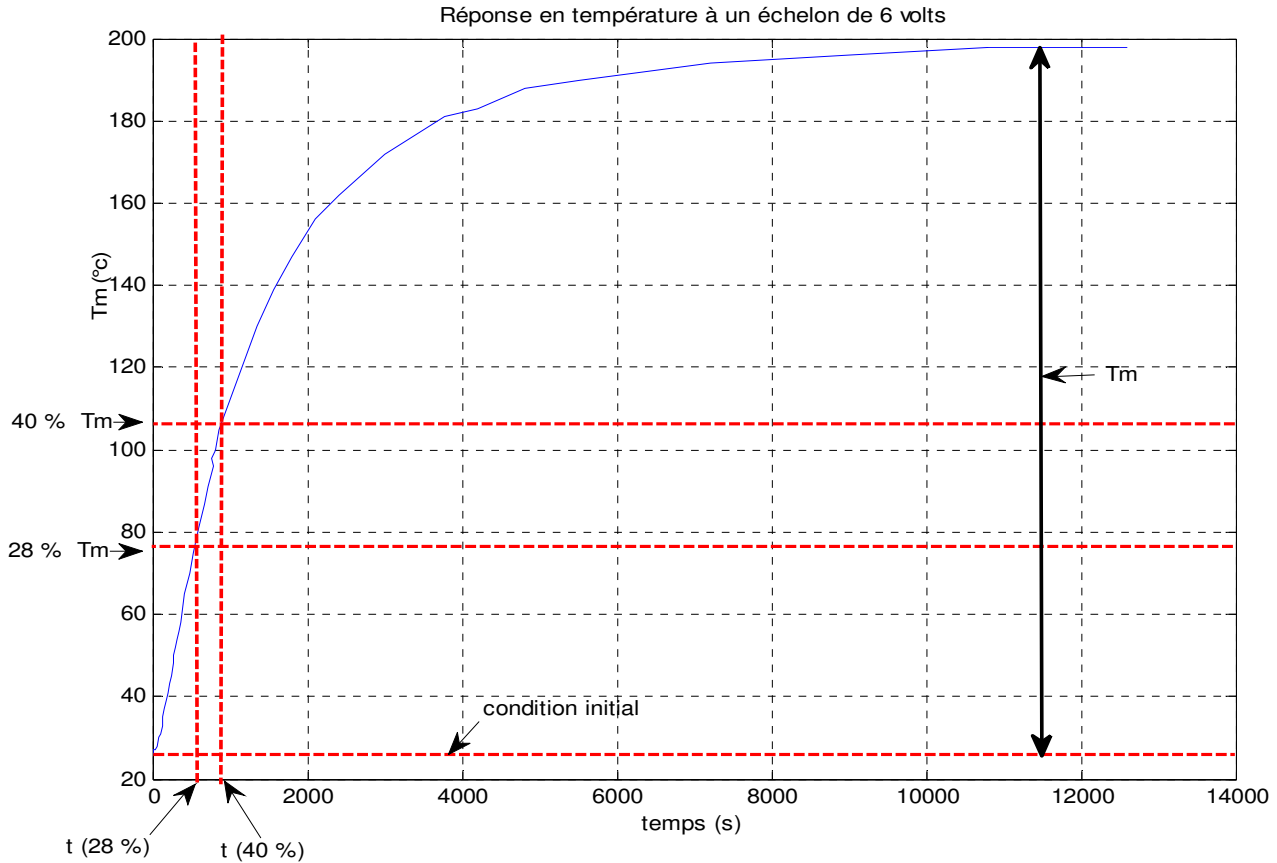


Figure I.7 : Réponse en température un échelon d'entrée de 6 volt

Application de la méthode de Broïda :

➤ Détermination de gain K

En régime permanent ($t \rightarrow \infty$) on a : $E \cdot K = s(t) - s(0) = \Delta s \rightarrow$

$$K = \frac{s(t) - s(0)}{E} \tag{I.12}$$

$$K = \frac{7.32 - 1.08}{6} \tag{I.13}$$

K=1.22

➤ Détermination de T et τ :

On doit d'abord trouver $t_{28\%}$ et $t_{40\%}$

$$s(t_{28\%}) = \Delta s * 0.28 + s(0) = 6.9 * 0.28 + 1.08 = 3.012 \tag{I.14}$$

$$s(t_{40\%}) = \Delta s * 0.40 + s(0) = 6.9 * 0.40 + 1.08 = 3.840 \tag{I.15}$$

A partir de la figure 5 on trouve :

$$t_{28\%} = 170 \text{ s}$$

$$t_{40\%} = 240 \text{ s}$$

$$T = 1375 \text{ s}$$

Calcul de retard τ

$$\tau = 2.8 * 500 - 1.8 * 750 \quad \tau = 50 \text{ s}$$

$$G(P) = \frac{1.22e^{-50P}}{1+1375P} \tag{I.16}$$

On peut l'approximer à la forme suivante

$$G(P) = \frac{1.22}{(1+1375P)(1+50P)} \tag{I.17}$$

La réponse indicielle de G(p) est illustrée sur la figure I.8

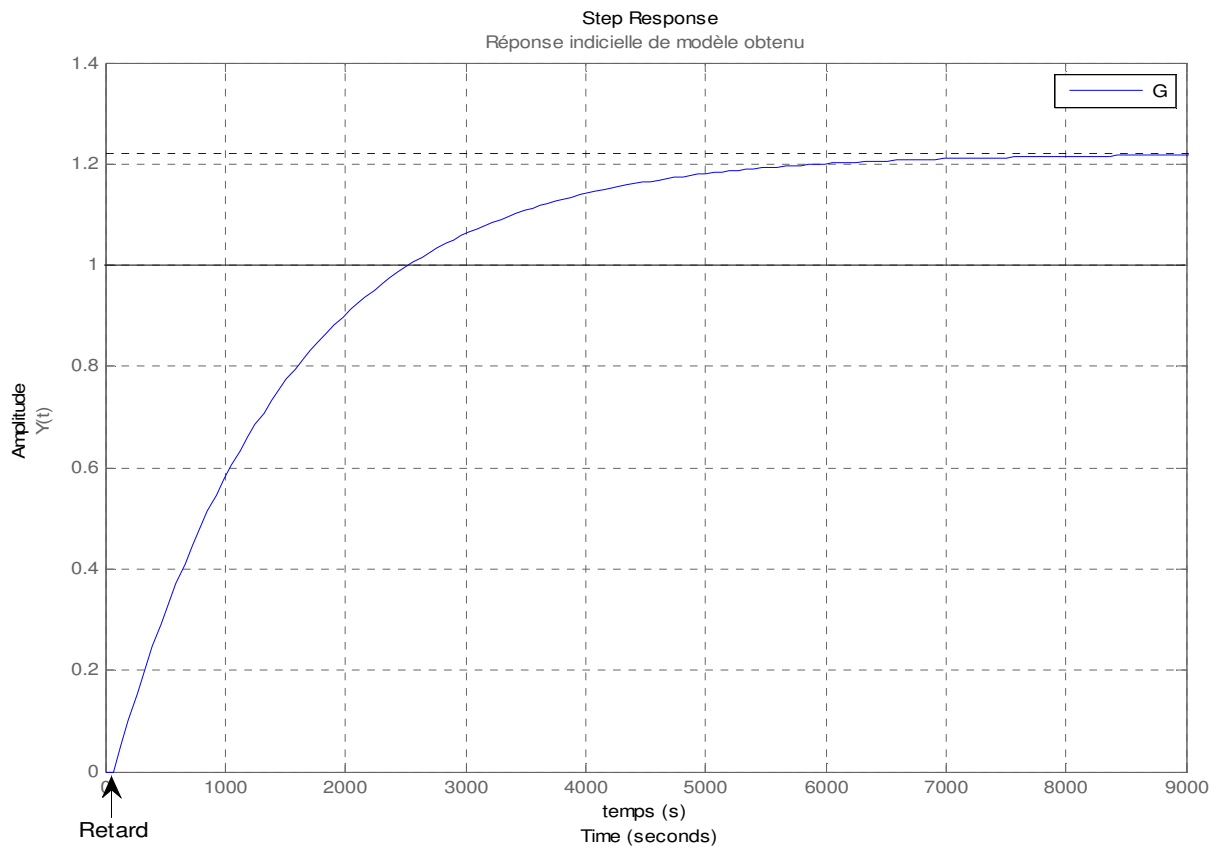


Figure I.8 : réponse indicielle de modèle obtenu

I.3 MODILISATION DU FOUR

Comme il est illustré sur la figure I.8, la chaleur à l'intérieur du four électrique, à modéliser, est produite par une résistance chauffante, commandé en tension V_c par un amplificateur de puissance. La mesure de la température se fait à partir d'un thermocouple placé dans une cavité de mesure et d'un amplificateur d'instrumentation produisant une tension V_m , image de la température θ_m . L'ensemble capteur et amplificateur d'instrumentation sont supposés linéaires dans la gamme de température du four. [3]

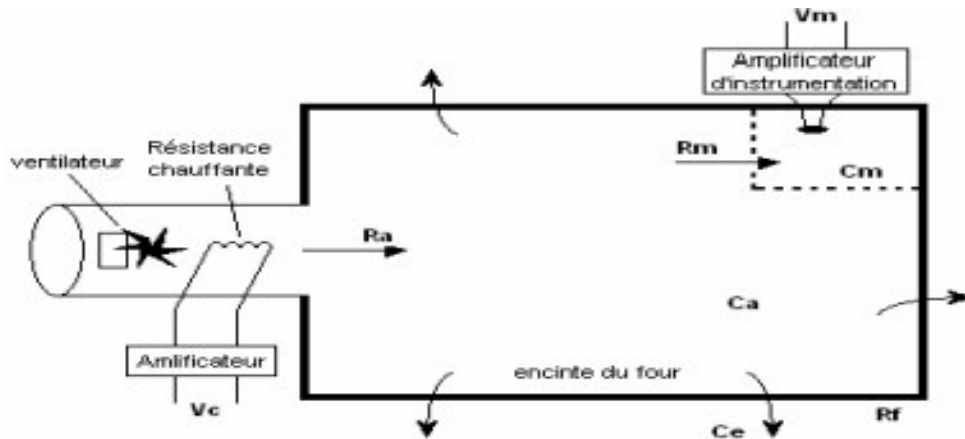


Figure I.8 : Processus d'un four ventilé

La figure I.9 montre le schéma électrique d'un four ventilé. Ce dernier est résistif-capacitif.

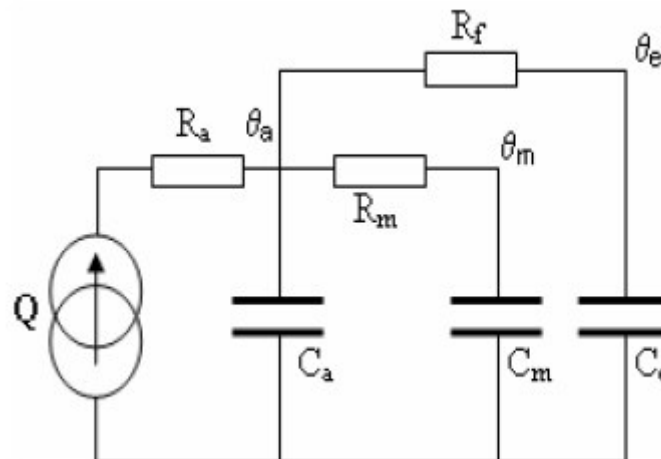


Figure I.9 Schéma électrique du four [1]

Les différents paramètres constituant ce processus sont

- $Q = k_1 V_c$: quantité de chaleur produite,
- R_a : résistance thermique freinant la circulation de la chaleur du conduit jusqu'à l'enceinte du four,
- C_a : capacité calorifique de l'enceinte du four,

- R_m : résistance thermique freinant la circulation de la chaleur du four à l'intérieur de la cavité de mesure,
- C_m : capacité calorifique de la cavité de mesure,
- R_f : résistance de fuite freinant la circulation de la chaleur vers l'extérieur du four,
- calorifique extérieur du four, considérée comme infinie,
- $\theta_a, \theta_m, \theta_e$: température respectives de l'enceinte du four, de la cavité de mesure, et e l'extérieur du four,
- θ_m : température mesurée son image est la tension V_m ($V_m = k_2 \theta_m$).

Par identification entre les paramètres du four et le schéma électrique et en considérant la quantité de chaleur Q comme étant le courant, on obtient les équations suivantes :

$$Q = C_a \frac{d\theta_a}{dt} + C_m \frac{d\theta_m}{dt} + \frac{\theta_a - \theta_e}{R_f} \tag{I.18}$$

Étant le courant qui traverse la capacité. La température θ_m est exprimée par l'équation suivante

$$\theta_m = \theta_a - R_m C_m \frac{d\theta_m}{dt} \tag{I.19}$$

Les paramètres θ_a, θ_m et θ_e ont été choisis comme étant respectivement les tensions (images des températures) des capacités C_a, C_m et C_e . En utilisant la transformée de Laplace des expression suivantes (I.18) et (I.19), nous obtenons les équations suivantes

$$\theta_a = \left(Q + \frac{\theta_e}{R_f} \right) \left(\frac{R_f(1+R_m C_m P)}{1+(R_m C_m + R_f C_m + R_f C_a)p + R_f R_m C_m C_a p^2} \right) \tag{I.19}$$

$$\frac{\theta_m}{\theta_a} = \frac{1}{1+R_m C_m P} \tag{I.20}$$

Compte tenu des équations (I.18) et (I.19), le schéma fonctionnel du processus complet peut être décrit comme suit:

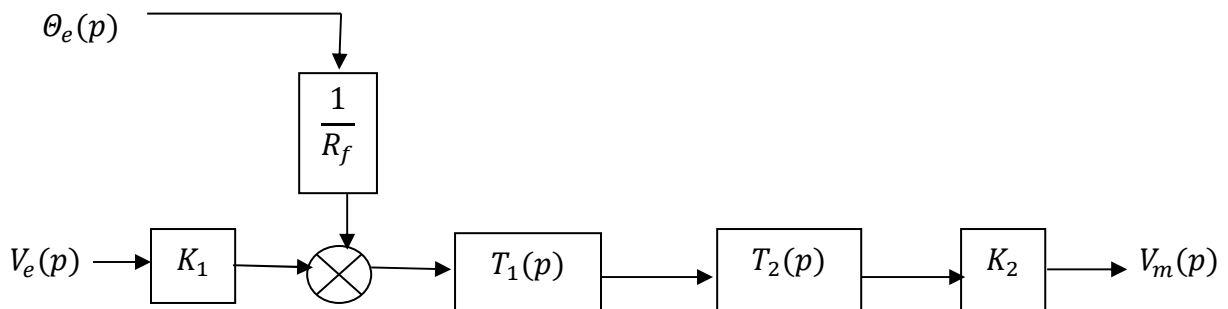


Figure I.10 : Schéma fonctionnel du four

Avec

$$T_1(p) = \frac{R_f(1+R_m C_m p)}{1+(R_m C_m + R_f C_m + R_f C_a)p + R_f R_m C_m C_a p^2} \quad (I.21)$$

$$T_2(p) = \frac{1}{1+R_m C_m p} \quad (I.22)$$

Les paramètres du modèle du four sont dressés sur le tableau I.2

R_a	0.01 °C / W	C_a	5000 J / °C
R_m	3 °C / W	C_m	10 J / °C
R_f	0.1 °C / W	C_e	∞
Q_{max}	5000 W	k_1	1 W / V
θ_e	20 °C	k_2	2.834 V / °C

Table I.2 : Paramètres du four

Après les calculs

$$T_1(p) = \frac{0.1+3p}{1+531p+1500p^2} \quad (I.23)$$

$$T_2(p) = \frac{1}{1+30p} \quad (I.24)$$

I.4 Etude comparative des modèles obtenu

Pour pouvoir comparer les deux modèles (modèle identifié et celui de la modélisation mathématique). Nous avons simulé les deux modèles dans le même bloc soumis à la même entrée échelon de 1volt, la figure suivante montre le schéma de simulation suivant :

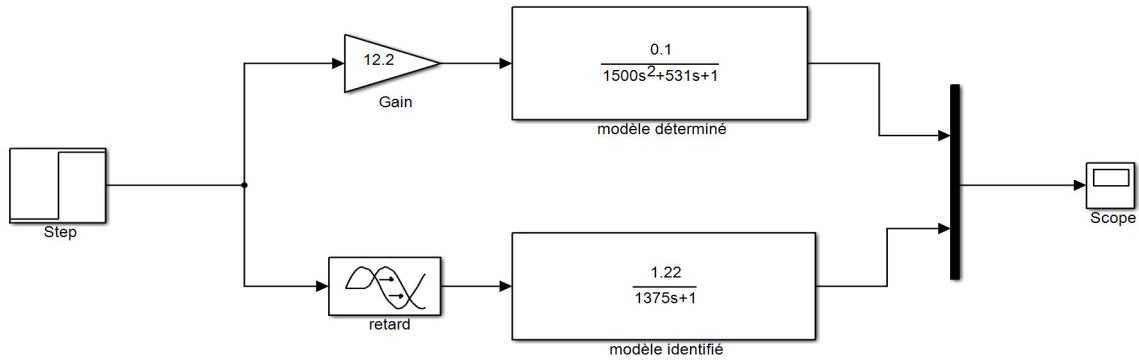


Figure II.11 : Modèle analogique déterminé et celui identifié au laboratoire

Les résultats de simulation sont donnés sur la figure II.12

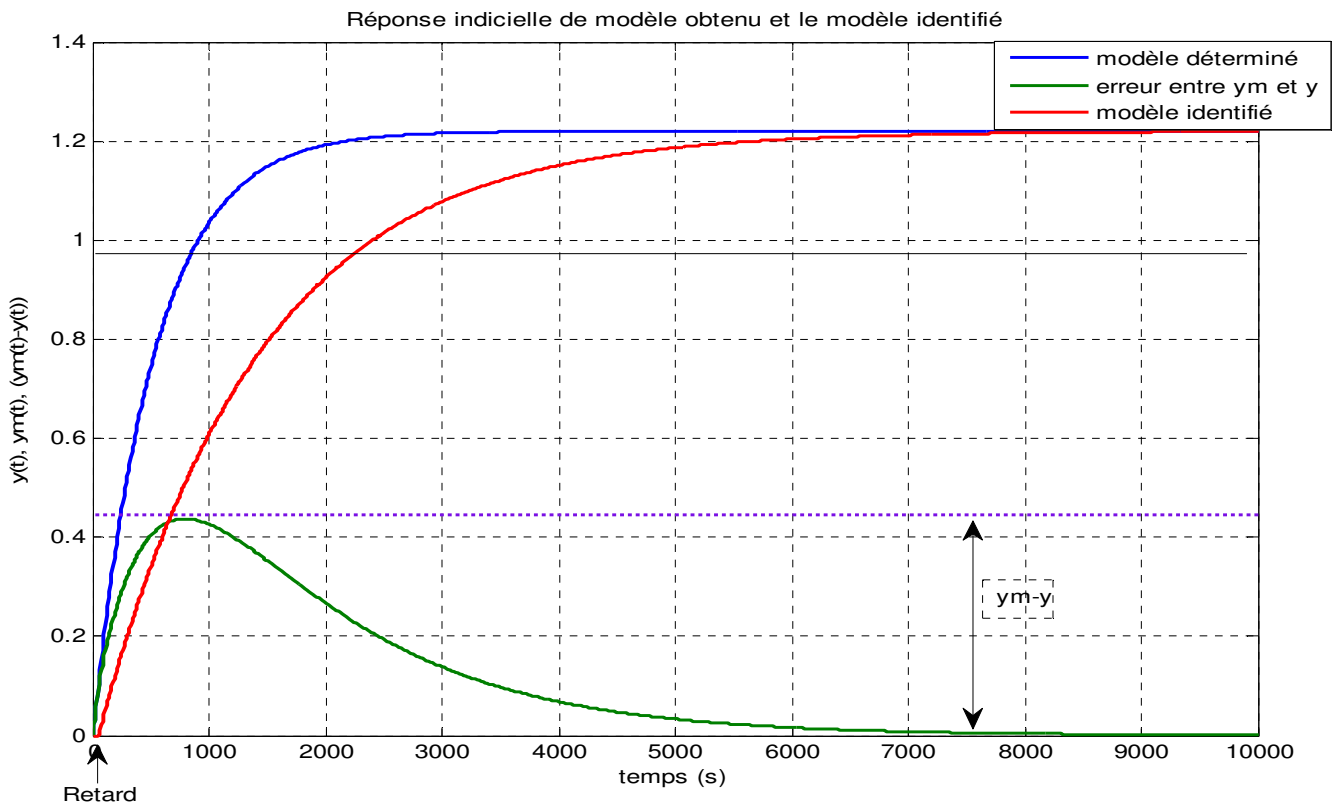


figure.II.12 : réponse à un échelon de 1 volt de deux modèles

Remarque :

On remarque une différence entre les deux réponses, le modèle déterminé est plus rapide que le modèle identifié ça due aux composantes différentes des deux fours.

I.5 Conclusion

Nous avons vus dans ce chapitre que le comportement du four est un première ordre avec un retard pur, après avoir obtenu le modèle mathématique de notre système, il nous reste de synthétiser un régulateur approprié, mais pour ce genre de système il nous faut une structure de réglage différente que la structure classique PID, dans le chapitre suivant nous allons voir comment et pourquoi une structure RST mieux adapté pour notre système.

Chapitre II

**Application de la Commande
RST pour la Régulation de
Température**

Application de la Commande RST pour la régulation de température

II.1 Introduction

Le développement extraordinaire des microprocesseurs a provoqué des changements importants dans la conception des systèmes de commande/régulation. Leur puissance de calcul et leur faible coût les rendent aptes à prendre également en charge les aspects commande/régulation avec des performances nettement supérieures à celle de régulateur analogique. Pratiquement tous les systèmes de commande / régulation construits actuellement utilisent des microprocesseurs. Pour tirer réellement profit des capacités des microprocesseurs, il ne suffit pas de reproduire le comportement des régulateurs PID analogiques, il faut mettre en œuvre des techniques d'automatique spécifiques et plus performantes, développées pour la commande par ordinateur, techniques testées industriellement ces vingt dernières années. Par ailleurs, le développement des logiciels d'automatique et de leur utilisation a permis d'améliorer d'une façon significative l'efficacité de la conception et de la mise en œuvre des systèmes de commande/régulation. La dissimulation des PC-machines ouvertes pouvait être facilement complétée avec des coûts d'acquisition qui ont permis d'une part la généralisation de l'usage des logiciels, et a conduit à d'autre part à la création des stations de travail pour l'analyse, la conception, le développement et la mise en œuvre sur des procédés réels de différents types d'algorithmes de commande/régulation.

Dans le présent travail, nous étudions en simulation l'application d'une commande numérique RST à un procédé thermique (four électrique). Cette commande a été calculée de manière à répondre à un cahier des charges qui fixe les performances désirées du procédé en boucle fermée.

II.2 Présentation de la commande RST

Le schéma de régulation utilisé correspond à la figure II.1 et permet une meilleure exploitation des possibilités de la commande numérique que l'utilisation d'un simple PID, avec une approche permettant une mise en œuvre aisée sur calculateur numérique (automates, microcontrôleurs etc.). [4]

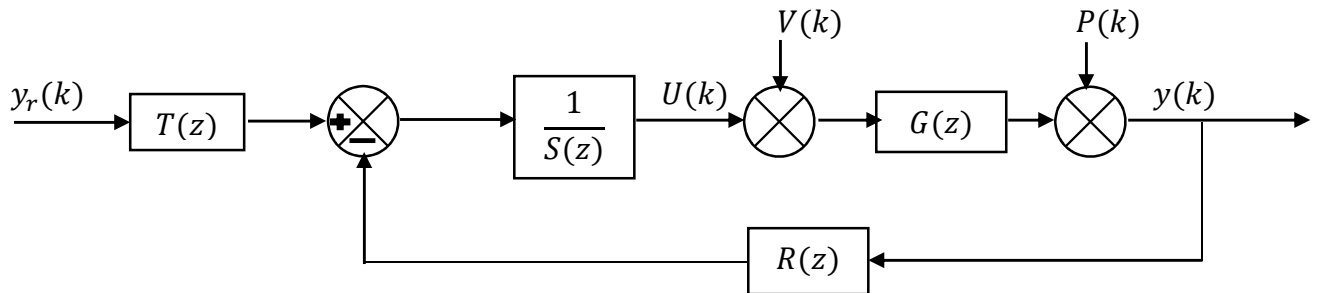


Figure II.1 structure générale du régulateur RST

II.2.1 description des éléments de la boucle de régulation

$y_r(k)$ est la consigne.

$v(k)$ est la perturbation agissant au niveau de la commande.

$P(k)$ est la perturbation agissant au niveau de la sortie de procédé

$y(k)$ est la sortie de procédé

$G(z)$ fonction de transfert discrète du procédé

$u(k)$ la commande appliqué au procédé

$R(z)$, $S(z)$ et $T(z)$ sont les polynômes de régulateur

II.2.2 avantages de régulateur RST

Le régulateur possède deux degrés de liberté : le premier classiquement défini sur le signal d'erreur consigne/mesure, le second autorisant la poursuite via une trajectoire de référence. La structure du contrôleur permet aussi d'imposer des pôles et certains zéros en boucle fermée. [4]

La plupart des boucles d'asservissement mettent en œuvre le classique correcteur PID.

Bien que relativement facile à régler, le PID ne permet pas toujours d'obtenir les performances requises, en particulier sur les systèmes :

- à retard pur / temps mort important

- dont les caractéristiques dynamiques varient au cours du fonctionnement
- d'ordre supérieur à 2 (donc possédant plus d'un mode vibratoire)
- possédant des zones de non-linéarités (frottements, jeu ...)

Le correcteur numérique RST (dont le PID est un cas particulier) est bien adapté à ces types de systèmes. Il permet :

- de gérer, sans module complémentaire, le retard pur
- de commander des systèmes possédant jusqu'à 5 modes vibratoires. (ordre 10)
- de gérer de manières distinctes les dynamiques de suivi de consigne et de rejet de perturbation.
- La consigne peut être atteinte selon une trajectoire du 2nd ordre entièrement paramétrable (dépassement, temps de montée)
- d'obtenir une excellente robustesse du réglage, pour s'affranchir des variations dans la dynamique du système (gain, retard, inertie)

Il s'intègre sur toute cible programmable (carte contrôleur, automate programmable, PC ...)

Sa mise en œuvre s'effectue à partir d'un modèle paramétrique de la boucle et des spécifications des performances à atteindre : dépassement et temps de montée en suivi de consigne et réjection de perturbation.

II.3 Méthode de synthèse d'un régulateur RST

II.3.1 Principe

Un régulateur RST est un organe de contrôle permettant d'effectuer une régulation en boucle fermée d'un dispositif industriel. C'est un correcteur fréquemment utilisé dans les dispositifs de commande numérique.

Le sigle RST vient du nom des 3 polynômes devant être déterminés afin d'obtenir une commande efficace. La synthèse de ce type de correcteur s'effectue par placement de pôles. La résolution du dispositif met en œuvre un polynôme de poursuite, et un polynôme de filtrage.

Ces correcteurs permettent de réaliser un asservissement d'un processus de type P3. D'une structure plus complexe que la structure classique, ces régulateurs sont aussi plus souples. Le calcul de ces correcteurs peut se faire automatiquement par un logiciel. [5]

II.3.2 Différents types de système

Soit un système décrit par la fonction de transfert $F(z) = \frac{N^p}{D^q}$, avec p le degré de numérateur et q est le degré dominateur

II.3.2.1 processus de type P_1 :

Si les zéros sont stables et $p=q$ ou $p=q-1$ alors le processus est de type P_1 . Ce type est le plus facile à réguler. Pour ce type de processus c'est la méthode de la boucle ouverte qui convient en régulation

II.3.2.2 processus de type P_2

Si les zéros sont stable et $q-p \geq 2$, le processus est dit de type P_2 , pour la régulation de ce type de processus, on utilise compensation ou placement de pôles c'est-à-dire on se fixe un modèle désirée à atteindre. Pour ce type on utilise la méthode de la boucle fermée le système désiré est souvent de 2^{eme} ordre (ce que on 'appelé aussi méthode de Zdan).

II.3.2.3 processus de type P_3

S'il existe des zéros instables ($|z_0| > 1$) et $q-p$ est quelconque, et pour ce type de système un correcteur RST est mieux adapté pour lui.

Nous présentons ici une méthode de calcul de ce type de correcteur. On adopte une démarche en boucle fermée c'est à dire que l'on se fixe la fonction de transfert à atteindre en BF. Notons ce modèle :

$$H_m(z) = \frac{B_m(z)}{P(z)} \tag{II.1}$$

L'ensemble du processus, compensé par le régulateur RST, a pour fonction de transfert en boucle fermée la forme suivante :

$$\frac{Y(z)}{Y_r(z)} = \frac{B(z)T(z)}{A(z)S(z)+B(z)R(z)} \tag{II.2}$$

Les trois polynômes $R(z)$, $S(z)$ et $T(z)$ doivent être choisis et dimensionnés afin que cette fonction de transfert en boucle fermée réponde à un modèle de référence, ou modèle à poursuivre, préconisé par l'utilisateur équation (II.1).

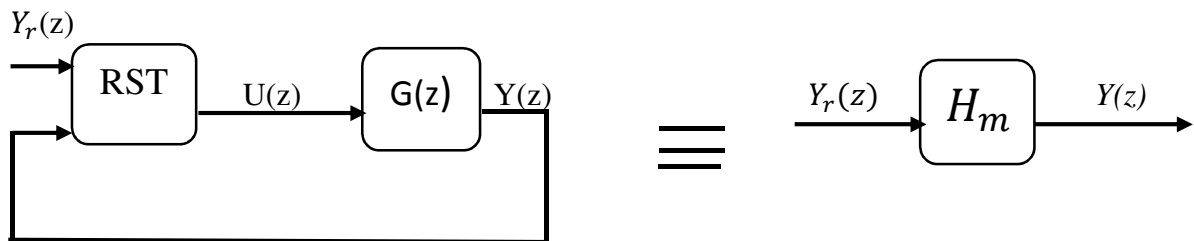


Figure II.2 : Principe de la synthèse algébrique d'un régulateur RST

Il est évident que, pour assurer la stabilité de l'installation, le polynôme $P(z)$ sera choisi pour ne présenter que des racines [pôles de $H_m(z)$] dont les points représentatifs sont tous contenus à l'intérieur du cercle de rayon-unité.

Le degré du polynôme $P(z)$ n'est pas forcément égal à celui de $A(z)R(z) + B(z)S(z)$. En fait, un modèle à poursuivre très simple, avec un polynôme $P(z)$ de degré nettement inférieur est généralement requis. [6]

Remarque : Le régulateur RST étant à deux degrés de liberté, les zéros en boucle fermée [i.e. les zéros de $B_m(z)$] peuvent eux aussi être positionnés dans le plan complexe, selon les désirs du concepteur.

II.3.3 Synthèse de R, S et T

Soit la fonction de transfert du système à régler

$$G(z) = \frac{B(z)}{A(z)} \tag{II.3}$$

Avec

$$B(z) = b_n z^n + b_{n-1} z^{n-1} + \dots + b_2 z^2 + b_1 z + b_0 \tag{II.4}$$

$$A(z) = z^n + a_{n-1} z^{n-1} + \dots + a_2 z^2 + a_1 z + a_0 \tag{II.5}$$

Nous fixons une fonction de transfert désirée $H_m(z)$, qui décrit les performances à atteindre durant la régulation :

$$H_m(z) = \frac{B_m(z)}{P(z)}$$

Avec

$$B_m(z) = b_{2n+1} z^{2n+1} + b_{2n} z^{2n} + \dots + b_1 z + b_0 \tag{II.6}$$

$$P(z) = P_{2n+1} z^{2n+1} + P_{2n} z^{2n} + \dots + P_1 z + P_0 \tag{II.7}$$

En boucle fermée, la fonction de transfert du système du régulateur RST à pour expression :

$$G_{BF}(z) = \frac{B(z)T(z)}{A(z)S(z)+B(z)R(z)} = \frac{B_{BF}(z)}{A_{BF}(z)} = \frac{B_m(z)}{P(z)} = H_m(z) \tag{II.8}$$

- Le polynôme $A_{BF}(z)$ contient et définit les pôles désirés en boucle fermée. On les obtient en ajustant les paramètres du régulateur, lesquels sont reliés directement aux performances en régulation (pulsation propre, dépassement, constante du temps...).
- Le numérateur $B_{BF}(z)$, définit les zéros du système en boucle fermée

Les polynômes R, S et T sont donnés par les expressions suivantes

$$R(z) = r_n z^n + r_{n-1} z^{n-1} + \dots + r_1 z + r_0 \tag{II.9}$$

$$S(z) = s_{n+1} z^{n+1} + s_n z^n + s_{n-1} z^{n-1} + \dots + s_1 z + s_0 \tag{II.10}$$

$$T(z) = t_{n+1} z^{n+1} + t_n z^n + \dots + t_0 \tag{II.11}$$

Le problème consiste à déterminer les constantes

$r_0, r_1, r_2, \dots, r_n ; s_0, s_1, s_2, \dots, s_{n+1}$ et $t_0, t_1, t_2, \dots, t_h$

Pour calculer ces trois polynômes on fait appel à l'équation (II.8), par identification on a :

$$B(z)T(z) = B_m(z) \implies T(z) = \frac{B_m(z)}{B(z)} \tag{II.12}$$

Il est bien appaait que T(z) est simple à calculer car B_m(z) et B(z) sont connus.

Calcul des polynômes R(z) et S(z) à partir de l'égalité (II.8)

Le calcul des polynômes R(z) et S(z) reviens à résoudre l'équation de Bézout ou diophantine suivante :

$$A(z)S(z) + B(z)R(z) = P(z) \implies$$

$$(z^n + a_{n-1}z^{n-1} + \dots + a_2z^2 + a_1z + a_0)(s_{n+1}z^{n+1} + s_nz^n + s_{n-1}z^{n-1} + \dots + s_1z + s_0) + (b_nz^n + b_{n-1}z^{n-1} + \dots + b_2z^2 + b_1z + b_0)(r_nz^n + r_{n-1}z^{n-1} + \dots + r_1z + r_0) = P_{2n+1}z^{2n+1} + P_{2n}z^{2n} + \dots + P_1z + P_0 \tag{II.13}$$

Après développement de cette égalité on obtient la forme suivante [7]

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ a_{n-1} & \ddots & 0 & 0 & b_n & & & & & 0 \\ a_{n-2} & \ddots & \ddots & \ddots & 0 & b_{n-1} & & & & 0 \\ \vdots & \ddots & \ddots & \ddots & 0 & \vdots & \ddots & \ddots & & 0 \\ a_0 & \ddots & \ddots & 1 & b_1 & \ddots & \ddots & & & 0 \\ 0 & a_0 & 0 & \ddots & a_{n-1} & b_0 & \ddots & \ddots & b_n & \\ 0 & 0 & \ddots & \ddots & a_{n-1} & 0 & \ddots & \ddots & \vdots & b_{n-1} \\ 0 & & \ddots & \ddots & \vdots & 0 & \ddots & \ddots & \vdots & \vdots \\ 0 & & & 0 & a_0 & 0 & & 0 & \ddots & b_1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & b_0 \end{bmatrix} \begin{bmatrix} s_{n+1} \\ s_n \\ s_{n-1} \\ \vdots \\ s_1 \\ r_n \\ r_{n-1} \\ r_{n-2} \\ \vdots \\ r_0 \end{bmatrix} = \begin{bmatrix} p_{2n+1} \\ p_{2n} \\ p_{2n-1} \\ \vdots \\ p_n \\ p_{n+1} \\ p_{n+2} \\ p_{n+3} \\ \vdots \\ p_0 \end{bmatrix} \tag{II.14}$$

Ce système linéaire où les inconnues sont les s_i et les r_i est appelé système de Sylvester.

Ce système est inversible que si les polynômes A(z) et B(z) sont premiers entre eux (A ∧ B=1).

II.4 Application de la commande RST pour la régulation de température

On prend la fonction de transfert du four électrique identifié dans la chapitre précédent

$$\text{avec } G(P) = \frac{1.22.e^{-72P}}{1+1430P}$$

La transformé en Z de G(p) donne la forme suivante

$$G(z) = Z^{-3} \frac{0.01526 z + 0.01007}{z - 0.9792} \quad (\text{II.14})$$

Avec z^{-1} représente le retard

On tire de G(z) les 2 polynômes B(z) et A(z)

$$B(z) = 0.01526 z + 0.01007 \quad (\text{II.15})$$

$$A(z) = z - 0.9792 \quad (\text{II.16})$$

$$\text{Soit la fonction de transfert désirée } H_m(p) = \frac{1}{1+900p} \quad (\text{II.17})$$

$$\text{La transformé en Z de (II.17) donne } H_m(z) = \frac{0.03278}{z - 0.9792} \quad (\text{II.18})$$

$$\text{alors } B_m(z) = 0.03921 \quad \text{et} \quad P(z) = 1 - 0.9608z^{-1}$$

La réponse à un échelon unitaire de G(p) et $H_m(p)$ avant de placer le régulateur est illustrée sur la figure II.3

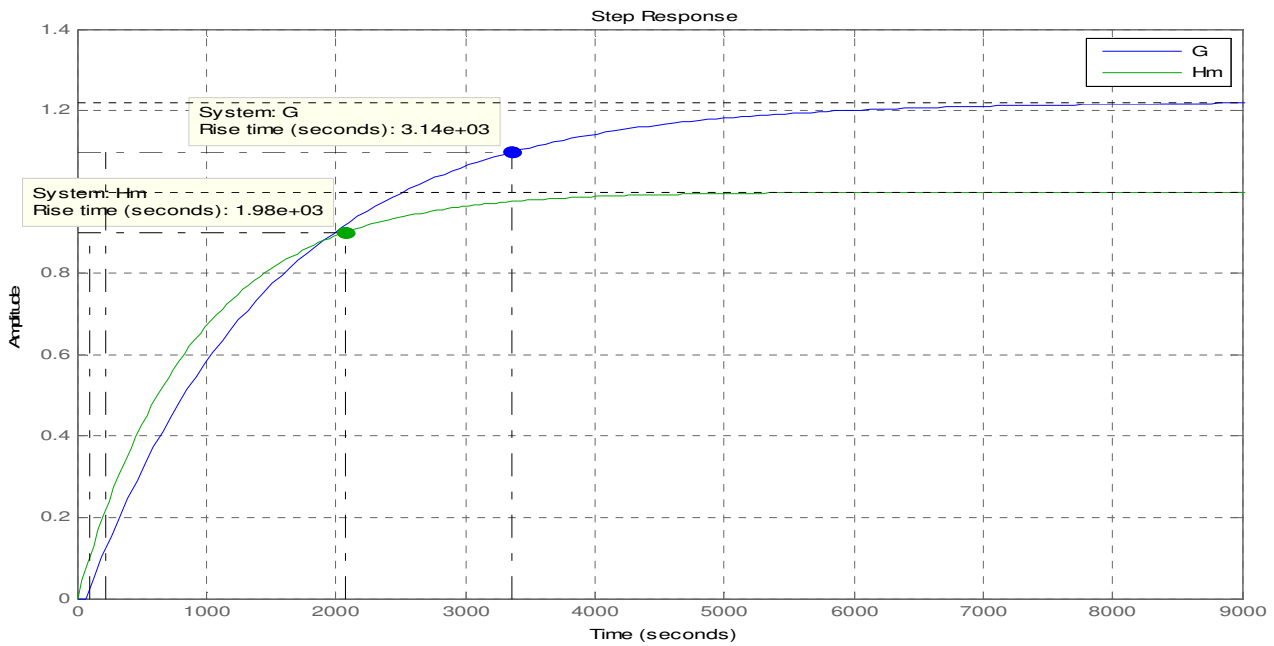


Figure II.3 : Réponse indicielle de modèle désiré et modèle identifié

La fonction de transfert en boucle fermée de système régulateur est comme suivant

$$G_{BF}(z) = z^{-3} \frac{B(z)T(z)}{A(z)S(z)+z^{-3}B(z)R(z)} = \frac{B_m(z)}{P(z)}$$

Calcul de T(z) :

de l'égalité (II.12) on peut tirer directement le T(z)

$$T(z) = \frac{2.148 z^2}{z + 0.6599} \tag{II.19}$$

Le Calcul R et S revient à la résolution de l'équation de Bezout et cette dernière a une unique solution ssi

$$\left\{ \begin{array}{l} \deg P \leq \deg A + \deg B + \text{retard} - 1 \\ \deg S = \deg B + \text{retard} - 1 \\ \deg R = \deg A - 1 \end{array} \right.$$

Vérification des degrés :

$$\left\{ \begin{array}{l} 1 < 1 + 1 + 3 - 1 \text{ est vérifié} \\ \deg S = 1 + 3 - 1 = 3 \\ \deg R = 1 - 1 = 0 \end{array} \right.$$

Après avoir résolu le système Sylvester nous obtenons les polynômes R et S

$$R(z) = 0.4600 \tag{II.20}$$

$$S(z) = \frac{z^3}{z^3 + 0.012z^2 + 0.004731z} \tag{II.21}$$

La réponse de système en boucle fermée sur les perturbations est donnée par la figure II.4

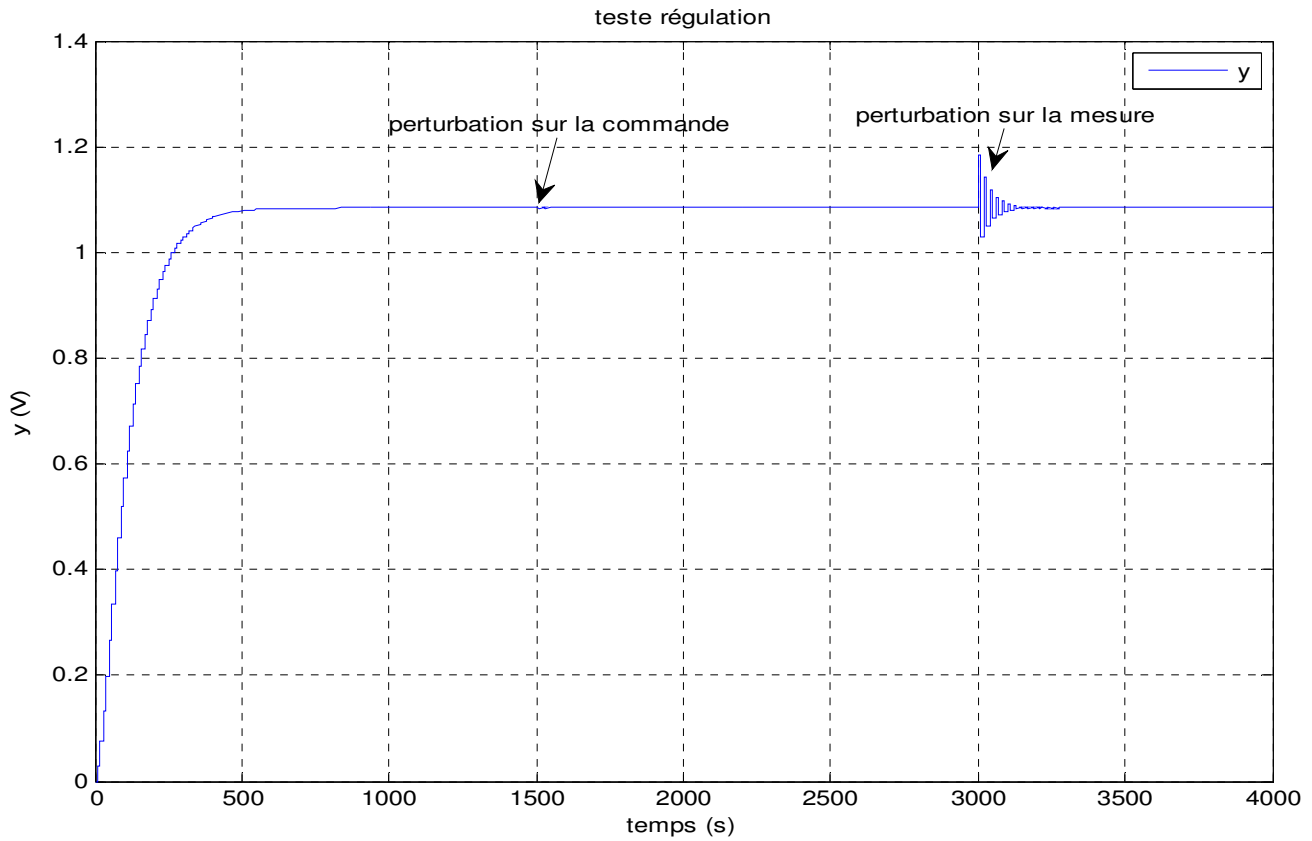


Figure II.4 : test régulation

Nous appliquons une consigne de 5 volt après 2000 seconde nous la changeons à 7 volt, les résultats de simulation sont présentés sur la figure II.5

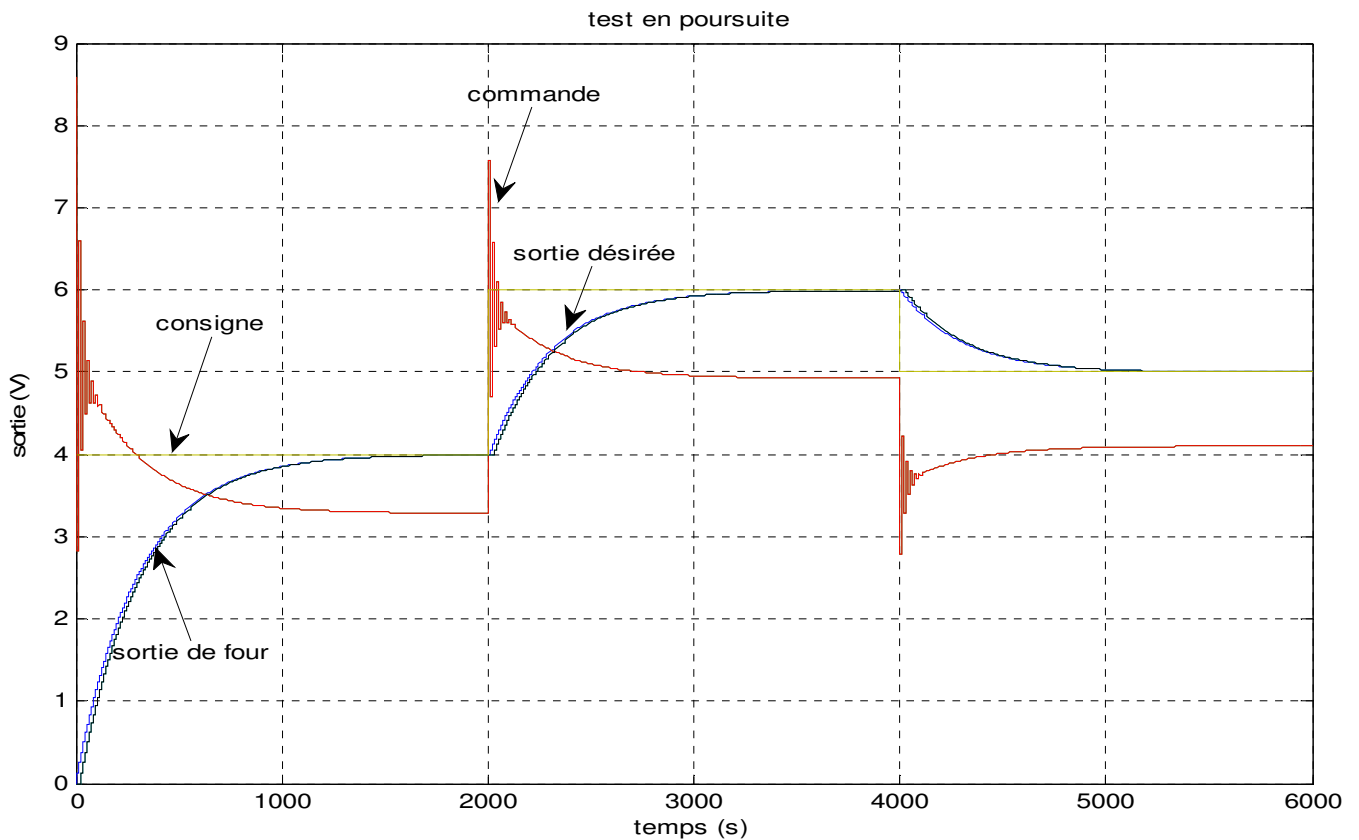


Figure II.5 : Réponse de système en poursuite

II.5 Conclusion

Dans ce chapitre nous avons exploités la méthode de synthèse d'un régulateur RST. Par la suite nous avons synthétisés régulateur RST pour la commande du four. En fin des tests de simulation ont été effectués afin de valider les résultats obtenus.

le chapitre suivant faire l'objet d'implémenter du régulateur obtenu sur un API S7-300.

Chapitre III

Implémentation de la Commande RST sur un API S7-300

Implémentation de la commande RST sur un API S7-300

III.1 Introduction

Les Automates Programmables Industriels (API) sont apparus aux Etats Unis vers 1969 où ils répondaient aux désirs des industries de l'automobile de développer des chaînes de fabrication automatisées qui pourraient suivre l'évolution des techniques et des modèles fabriqués. Un Automate Programmable Industriel (API) est une machine électronique programmable par un personnel non informaticien et destiné à piloter en ambiance industrielle et en temps réel des procédés industriels. Un automate programmable est adaptable à un maximum d'application, d'un point de vue traitement, composants, langage. C'est pour cela qu'il est de construction modulaire.

III.2 Présentation de l'automate programmable industrielle S7-300

La figure III.1 représente l'automate S7-300 de siemens utilisé pour notre projet



Figure III.1: API S7-300 CPU 314 2DP

III.2.1 Domaine d'application

La CPU 314C-2 DP est la CPU compacte pour des installations avec structure décentralisée. Cette CPU compacte avec sa mémoire de travail étendue est idéale pour des applications moyennes. Les entrées/sorties TOR et analogiques intégrées permettent un couplage direct au processus, l'interface PROFIBUS DP maître/esclave permet le raccordement d'unités périphériques décentralisées. Par conséquent, la CPU 314C-2 DP ainsi que le périphérique décentralisé peuvent être utilisés pour un prétraitement rapide, ou encore en tant qu'automate de niveau supérieur avec un système de bus de terrain. [8]

Les fonctions technologiques intégrées offrent d'autres possibilités d'utilisation :

- Comptage
- Mesure de fréquence
- Mesure de la période
- Modulation de largeur d'impulsions
- Régulation PID
- Positionnement en boucle ouverte

L'automate S7-300 est constitué d'une alimentation, des modules d'entrée et des modules de sortie. A ceux-ci peuvent s'ajouter des processus de communication et des modules de fonctionnement qui se chargeront des fonctions spéciales, tel que la commande d'un moteur pas à pas exemple.

Comme tout API, l'automate S7-300 est un système d'automatisation modulaire offrant la gamme des modules suivants :

- Module d'alimentation (PS) avec un courant de 2A, 5A, ou 10A.
- Unité centrale (CPU 314).
- Modules coupleur IM servant d'extension pour un autre rack.
- Module de fonctionnement (FM) pour réaliser des tâches spéciales.
- processeur de communication (PC) pour la connexion en réseau.

III.2.2 Structure interne d'un Automate programmable

La figure III.2 montre la structure interne d'un automate programmable

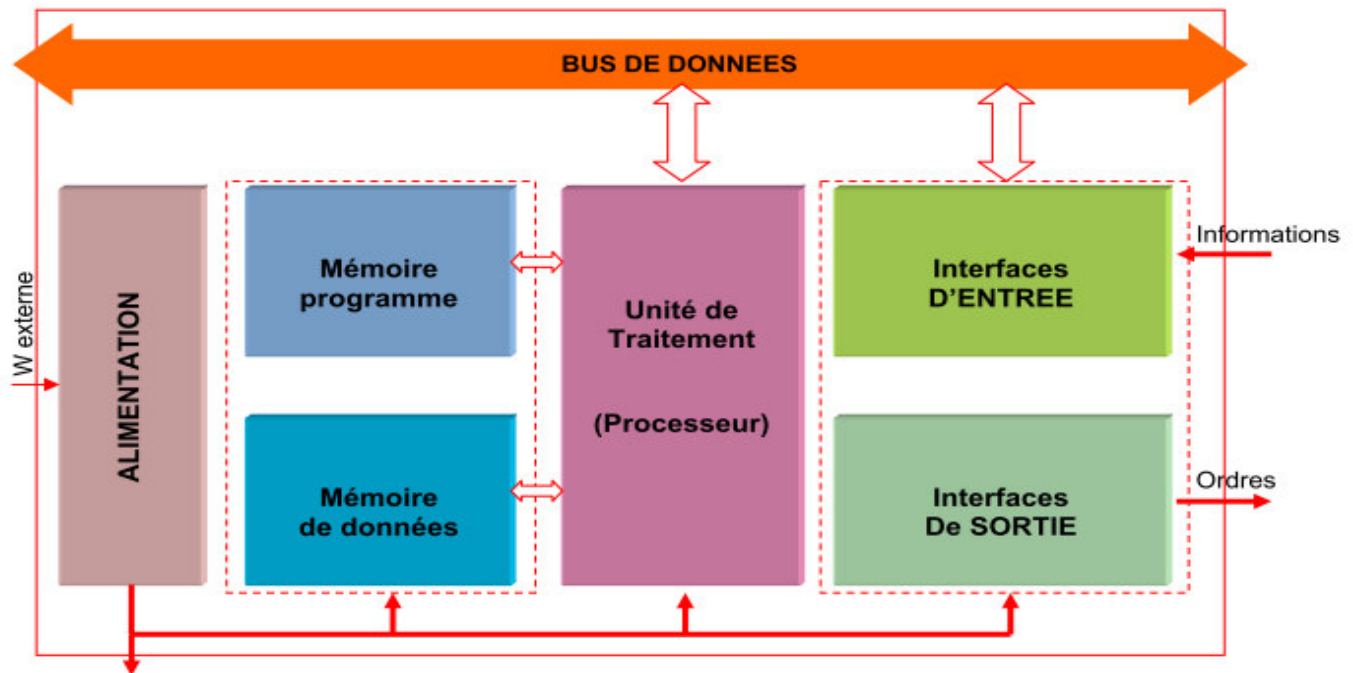


Figure III.1 : Structure interne d'un automate programmable

III.2.3 Caractéristiques de l'API S7-300

Le S7-300 présente les caractéristiques suivantes :

- Variété de la gamme de CPU.
- Différents types de modules permettant de réaliser des tâches d'automatisation plus efficaces.
- Possibilité d'exécution jusqu'à 32 modules.
- Possibilité d'appartenance en réseau avec les options **MPI**, **PROFIBUS**, **INDUSTRIAL ETHERNET**.
- Raccordement centrale de la PG avec accès à tous les modules.
- Configuration et paramétrage à l'aide de l'outil configuration matériel.

III.2.4 Différentes parties de l'API S7-300

III.2.4.1 Module d'alimentation

L'API S7-300 nécessite une tension de 24Vcc. Le module d'alimentation assure cette condition en transformant la tension du secteur 380/220Vca en 24Vcc. Il permet de sauvegarder le contenu de la RAM au moyen de la pile de sauvegarde ou d'une alimentation externe. Sur le module d'alimentation nous trouvons l'interrupteur de mise sous tension de l'automate. Tandis que, les tensions pour capteurs, actionneurs et voyants lumineux qui dépassent les 24 volts, sont fournies par des blocs d'alimentation ou transformateurs supplémentaires

III.2.4.2 Unité centrale

La CPU est le centre des opérations arithmétiques et logiques de l'automate. Elle reçoit les informations sur l'état des entrées, exécute le programme et met à jour les sorties. Elle permet le réglage du comportement au démarrage et le diagnostic de défauts par des voyants lumineux.

L'utilisateur peut choisir une des différentes gammes de la CPU offerte par le constructeur selon les performances désirées

III.2.4.3 Les module d'entrée/sortie

a. Module d'entrée

Il existe deux types d'entrées :

- Les modules d'entrées tout ou rien :

Un module d'entrées Tout ou Rien permet à l'unité centrale d'effectuer un lecteur de l'état logique des capteurs qui lui sont réalisés et de le matérialiser par un bit image de l'état du capteur.

- Les modules d'entrées analogiques permettant de :
 - Sélectionner la gamme d'entrées de chaque voie.
 - Scruter des voies d'entrées par le multiplexage et l'acquisition des valeurs.
 - Faire une conversion analogique/numérique des mesures d'entrées.

b. Modules de sortie

Il existe deux types de modules de sorties :

- Les modules de sortie Tout Ou Rien : ces modules permettent à l'automate programmable d'agir sur les prés-actionneurs ou de signaler à l'opérateur.
- Les modules de sortie analogique dont les différentes fonctions sont :
 - La sélection de la gamme pour chaque sortie
 - La conversion numérique/analogique

III.2.4.4 Les coupleurs

Les coupleurs permettent à l'automate de communiquer avec le milieu extérieur (console, imprimantes) ou de relier avec d'autres automates.

III.2.4.5 Modules de fonction (FM)

Ces modules permettent les fonctions suivantes le comptage, la régulation et le positionnement.

III.2.4.6 Fonctionnement de l'API

L'automate, pendant le fonctionnement exécute le programme cyclique, qui commence par l'acquisition des entrées issues de capteurs sur l'état du processus et se termine par l'envoi des sorties aux actionneurs.

Les informations peuvent être de type tout ou rien, analogique ou numérique.

III.3 Méthodologie d'implémentation sur API S7-300 [8]

Dans l'automatisation et de contrôle d'ingénierie, le logiciel MATLAB / Simulink de MathWorks est fréquemment utilisé pour modéliser graphiquement et simuler les processus et les contrôleurs.

L'exigence est d'exécuter les modèles Simulink créés en quelques étapes sur un contrôleur logique programmable.

MathWorks fournit la possibilité de compiler les modèles Simulink en C / C ++ en utilisant le code Embedded Coder. Ce code à son tour peut être intégré dans un projet WinAC ODK et appelé et exécuté à partir du programme S7 via DLL / RTDLL.

Cette application illustre l'intégration des modèles MATLAB / Simulink avec STEP 7 et WinAC ODK sur un automate S7_300.

WinAC cible facilite l'intégration et génère indépendamment:

- Fichier DLL ou RTDLL avec l'algorithme du modèle Simulink
- La source SCL avec les interfaces du modèle et la communication avec le fichier DLL / RTDLL

Cela nous permet rapidement et facilement de passer d'un modèle Simulink à un block fonctionnel dans step7 et de le charger dans l'automate S7_300.

En utilisant l'exemple d'un régulateur PID, nous fournissons une description détaillée de toutes les étapes: De la création du régulateur PID dans MATLAB / Simulink à l'exécution sur le contrôleur basé sur PC WinAC RTX ou bien sur l'automate. Il vous montre comment utiliser WinAC cible dans un gain de temps et de moyen facile d'intégrer Matlab/Simulink. Les modèles peuvent être intégrés pour STEP7 V5.5 et aussi pour STEP 7 V11 SP2 ou supérieur.

Pour intégrer le modèle Simulink, la figure III.3 nous montre les étapes à effectuer :

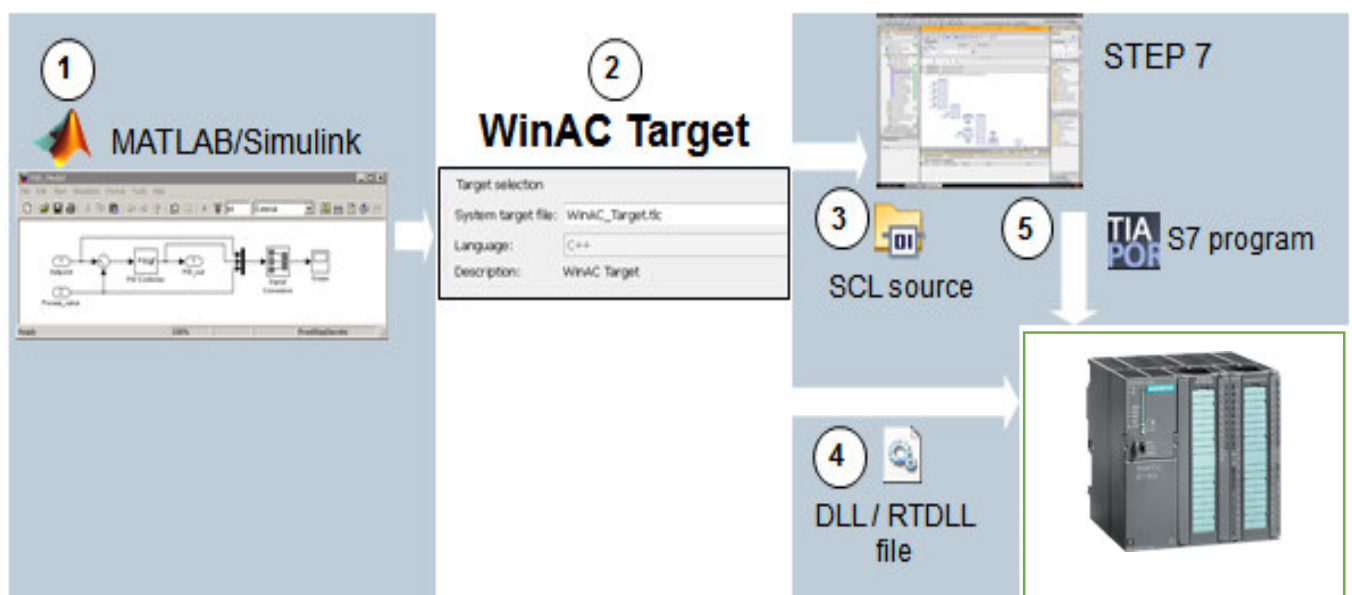


Figure III.3 : Etapes à suivre pour implémenter un régulateur PID

1. Création d'un modèle dans MATLAB / Simulink et de le tester dans MATLAB / Simulink

2. Compiler le modèle Simulink dans une source SCL et un fichier DLL / RTDLL utilisant WinAC cible
3. Intégrer la source SCL dans le programme S7
4. Copier le fichier DLL / RTDLL dans le disque C de notre ordinateur
5. Téléchargement du programme S7 dans l'automate et le lancer

Avantages [8]

Cette application fournit, en particulier, les avantages suivants:

- Facilité d'intégration de modèles Simulink dans Step7
- Mode externe Simulink pour la surveillance de modèles Simulink et en changeant les paramètres des modèles intégrés en cours de fonctionnement:
 - Voir en ligne graphique du processus en cours d'exécution
 - Modification des paramètres de modèles internes utilisant Simulink

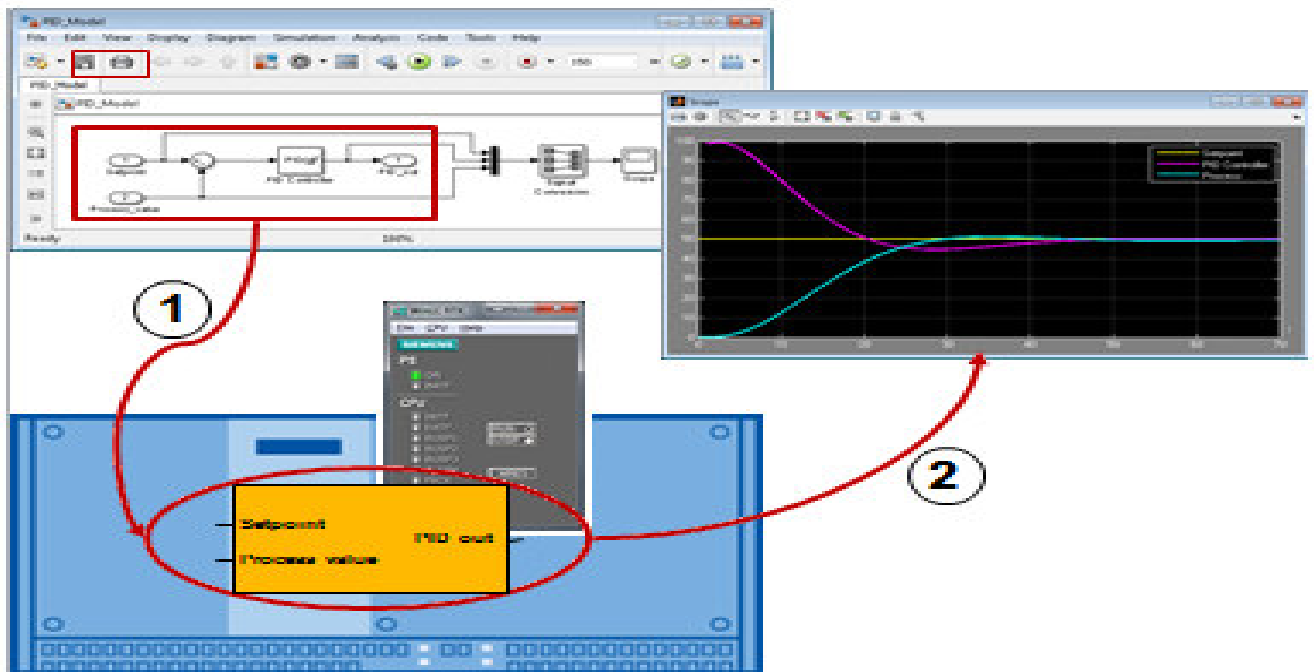


Figure III.4 : Schéma de visualisation des signaux

- Manipulation de paramètres du modèle interne dans le programme utilisateur avec STEP 7

Avantages

Cette application fournit, en particulier, les avantages suivants:

- Facile et rapide familiarisation avec WinAC cible à travers des instructions détaillées étape par étape
- Procédé optimisé pour l'intégration de modèles Simulink dans une cible programmable.
- La mise en service efficace et diagnostic avec Simulink et le mode externe
- Des options flexibles pour changer les paramètres du modèle internes en fonctionnement dans le programme STEP 7

III.4 Implémentation du régulateur RST pour la régulation de la Température

III.4.1 Création d'un modèle Simulink et génération d'une source SCL et une DLL Windows

Pour pouvoir générer le code C/C++ dont nous avons besoin puis le convertir en une source SCL et une DLL/RTDLL, nous suivons quelques étapes.

Dans le premier cas nous allons créer notre modèle Simulink avec le régulateur seul comme il est illustré sur schéma de la figure III.5

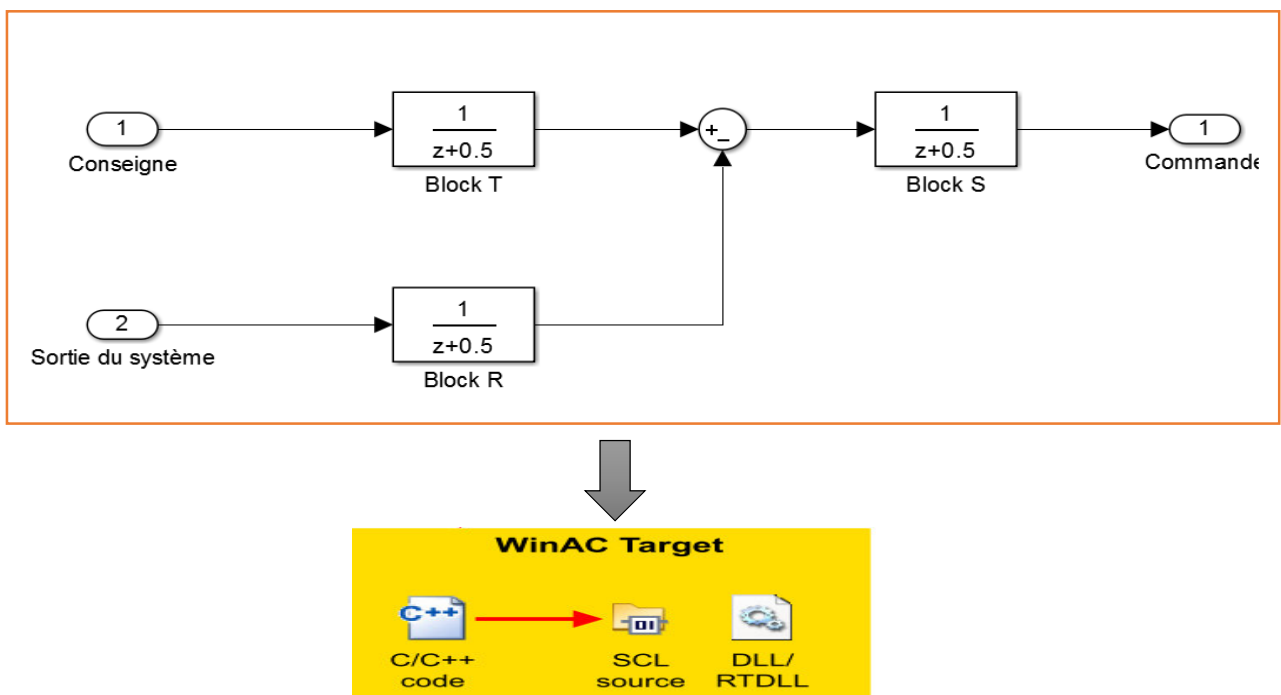


Figure III.5 : Modèle Simulink de régulateur RST

Avant de commencer la génération de code C/C++ nous devons faire quelques réglages dans Simulink le tableau III.1 nous montre les instructions à suivre

Étapes	Instruction
1	Ouvrez le modèle Simulink
2	Dans le menu, cliquez sur Code - C / C ++ code - Options de génération de code
3	<p>Une fenêtre avec une barre de navigation apparaît. Le poste «génération de code" est cochée par défaut.</p> <p>Définissez les paramètres suivants:</p> <p>"Génération de code" navigation item</p> <p>- Fichier cible du système: WinAC_Target.tlc</p>
4	<p>Les paramètres suivants seront réglés automatiquement après avoir sélectionné WinAC cible:</p> <p>"Solver" navigation item</p> <p>- Type: Fixe-étape</p> <p>- Solver: discrets (pas d'états continus)</p> <p>- Étape fixe la taille: auto</p> <p>"Interface" navigation item</p> <p>- L'échange de données</p> <ul style="list-style-type: none"> • Interface: Mode externe • couche transport: TCP-IP • arguments MEX-file: '192.168.0.2' <p>(Adresse IP du système d'exécution)</p> <p>"Data Import / Export" navigation item</p> <p>- Enregistrer dans l'espace de travail:</p> <p>Décochez les temps, Unis, la production, les états des variables finales</p>

	<p>"Options WinAC" navigation item</p> <ul style="list-style-type: none"> - Dans "WinAC ODK type de projet", vous pouvez choisir si vous voulez générer une DLL Windows (DLL) ou un temps réel DLL (RTDLL). UN Windows DLL est sélectionné par défaut. - Dans "STEP 7 Version", vous pouvez sélectionner le logiciel d'automatisation vous voulez utiliser plus tard. Le réglage de la V5.5 STEP 7 est sélectionné par défaut. - En "temps de cycle de données (mode Extended) [ms]", vous pouvez personnaliser le temps de cycle pour le mode externe. Le réglage par défaut est "50". - Dans "STEP 7 Fonction Nom du bloc", vous pouvez sélectionner le nom de la bloc de fonction. Le réglage par défaut est le nom du modèle Simulink. - Dans "L'accès aux paramètres avec STEP 7", vous pouvez activer le paramètre l'accès au modèle des paramètres avec le programme utilisateur STEP 7. - Dans "STEP 7 Bloc de données nom", vous pouvez sélectionner le nom des données bloc. Le réglage par défaut est "Model_Param_DB".
--	--

Tableau III.1 : les réglages nécessaires avant la génération du code

Remarque :

Lors de la sélection WinAC cible, le mode externe sera activé par défaut. Si vous ne voulez pas utiliser le mode externe, procédez comme suit:

- Dans le menu, cliquez sur "CODE - C / C ++ code - Options de génération de code ..."
- "Interface" navigation item
- Dans l'option de menu "échange de données", réglez "Interface" à "None"

Sortie WinAC cible

Pour compiler un modèle Simulink en code C ++ et de générer une DLL / RTDLL et une source SCL, nous procédons comme suit:

Etapas	Instruction
1	Dans le menu de modèle Simulink, cliquez sur “Code – C/C++ Code – Build Model” .
2	Dans ce cas, le bloc "RST_Control" du modèle Simulink (RST_Control.mdl) est compilé dans C / C ++. Puis WinAC cible génère la DLL/ RTDLL et la source SCL dont nous avons besoin.
3	<p>Le répertoire ... \ RST_Control_WinAC_Target_Output modèle de Simulink contient le code généré.</p> <p>Dans le fichier “Binary” nous trouvons la DLL/RTDLL</p> <p>Et dans le fichier “SCL_Source” nous trouvons la source SCL</p>

Tableau III.2 : Réglages pour la génération de code

Remarque :

Le texte « ### Successful completion of build procedure for model: RST_Control » dans l’espace de travail de Matlab nous montre que le codage a été effectué sans erreurs.

III.4.2 Implémentation de la source SCL et création de block « RST_Control » dans Step7

III.4.2.1 Intégration de la Bibliothèque WinAC ODK dans Step7

Pour intégrer des modèles, le logiciel WinAC ODK doit être installé. De plus, la Bibliothèque WinAC ODK doit être intégrée dans le SIMATIC Manager comme SFB65001 (CREA_COM) et SFB65002 (EXEC_COM) sont nécessaires pour exécuter la DLL / RTDLL. Leur rôle est :

- SFB65001 (CREA_COM) pour initialiser le fichier DLL / RTDLL
- SFB65002 (EXEC_COM) pour exécuter le fichier DLL / RTDLL

La figure III.6 nous montre comment intégrer cette bibliothèque dans le SIMATIC Manager.

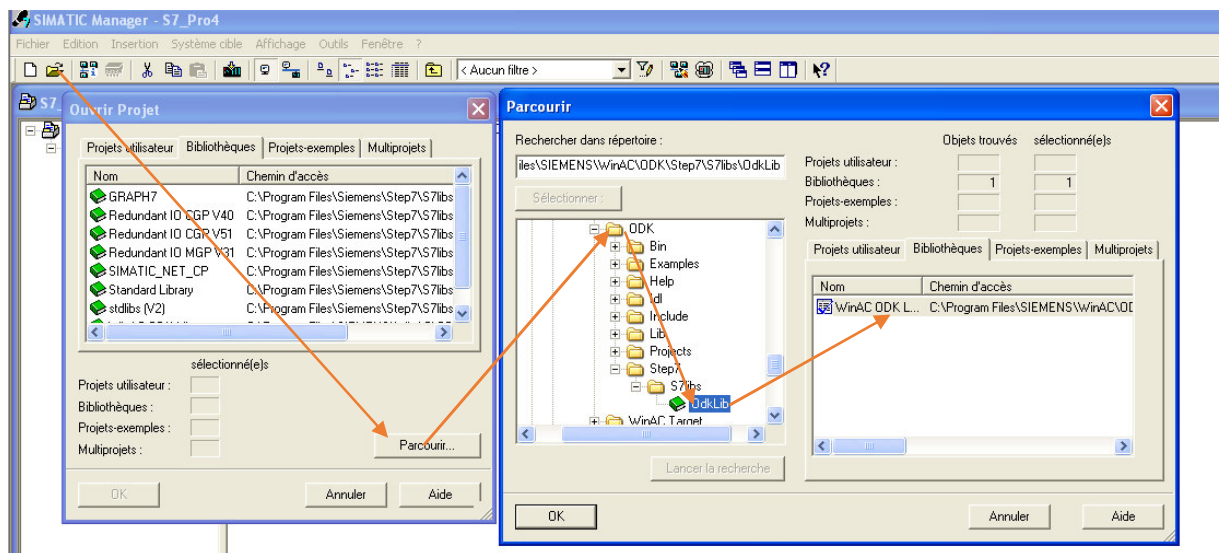
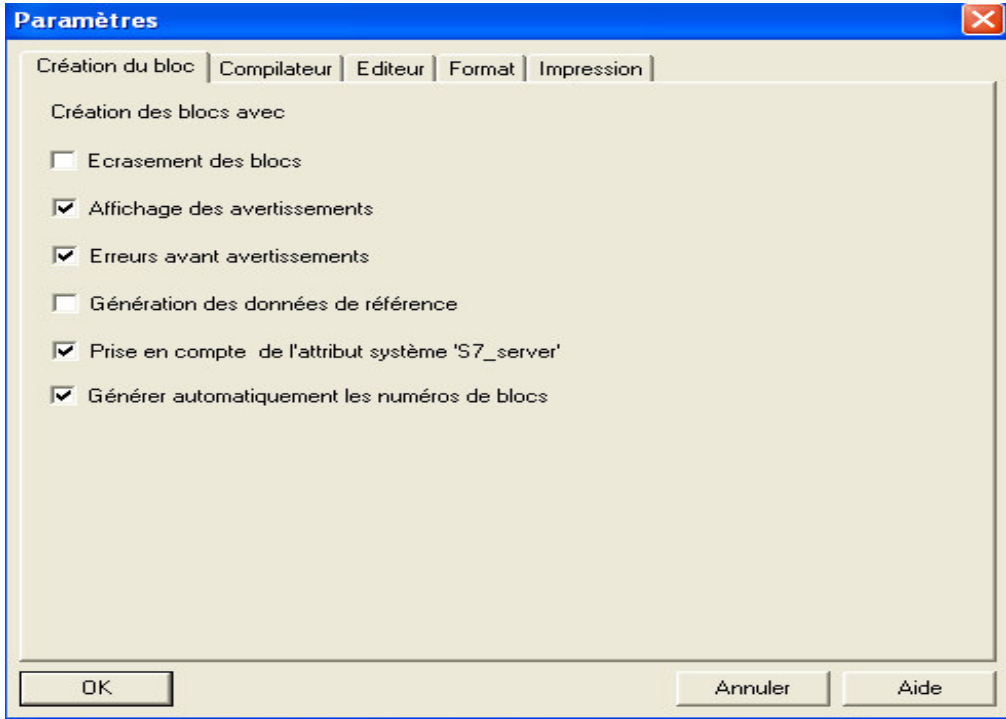


Figure III.6 : Intégration de la Bibliothèque WinAC ODK

Remarque : Il suffit de cliquer sur ok la bibliothèque sera intégrée dans notre SIMATIC Manager.

III.4.2.2 Création de block fonctionnel à partir de la source SCL

Le tableau III.3 nous montre les étapes à suivre pour la création de block fonctionnel qui contient notre régulateur

Etapes	Instruction
1	Ouvrez le SIMATIC Manager.
2	Ouvrez le projet préparé.
3	Aller dans le programme S7 et cliquer sur le dossier Source.
4	Ouvrez le dossier Source.
5	Ensuite, utilisez "Insérer - Source externe" pour ajouter la source SCL générée par WinAC cible.
6	Ouvrez la source SCL et cliquez dans le menu "Options - Personnaliser ..."
7	<p>Cocher "Créer automatiquement les numéros de blocs" et confirmez la fenêtre avec ok</p> 

8	Générer le bloc de fonction avec "Fichier - Compiler".
---	--

Tableau III.4 : les étapes de création de block fonctionnel qui contient le régulateur

La figure III.6 nous montre comment initialiser et exécuter la DLL avec OB100 et OB35

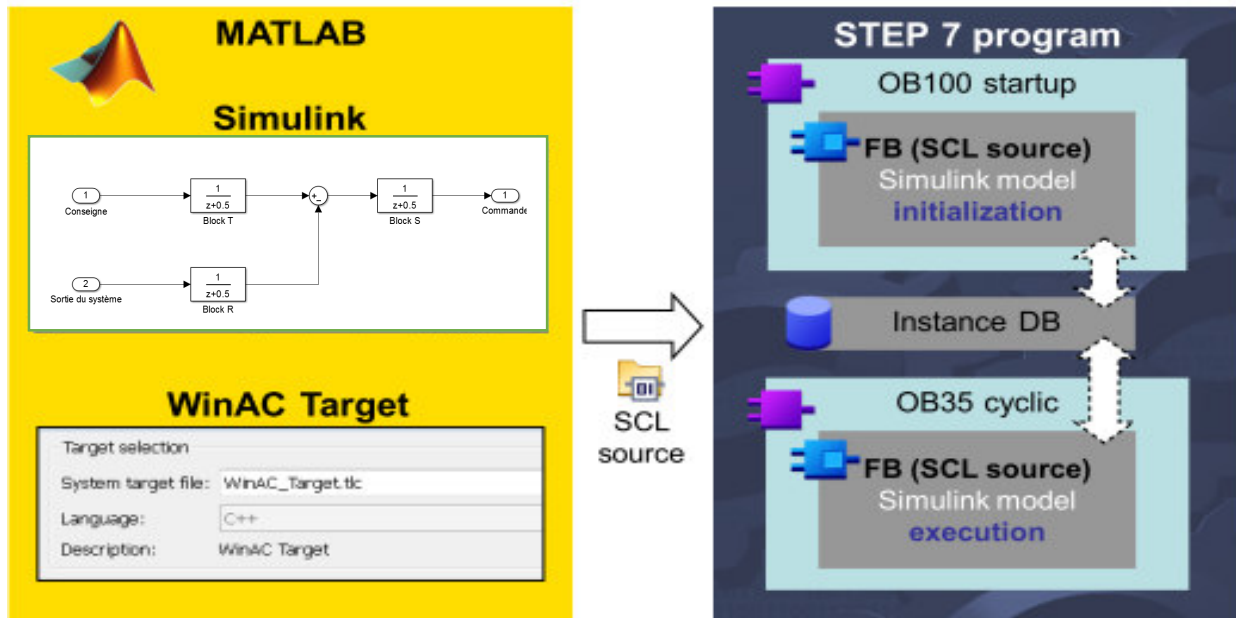


Figure III.6 : Modèle Simulink dans le programme STEP 7

Nous ouvrons le OB 100 et dans le Blocks FB nous trouvons notre régulateur sous nom « RST_Control » la figure III.8 nous montre la création de notre régulateur

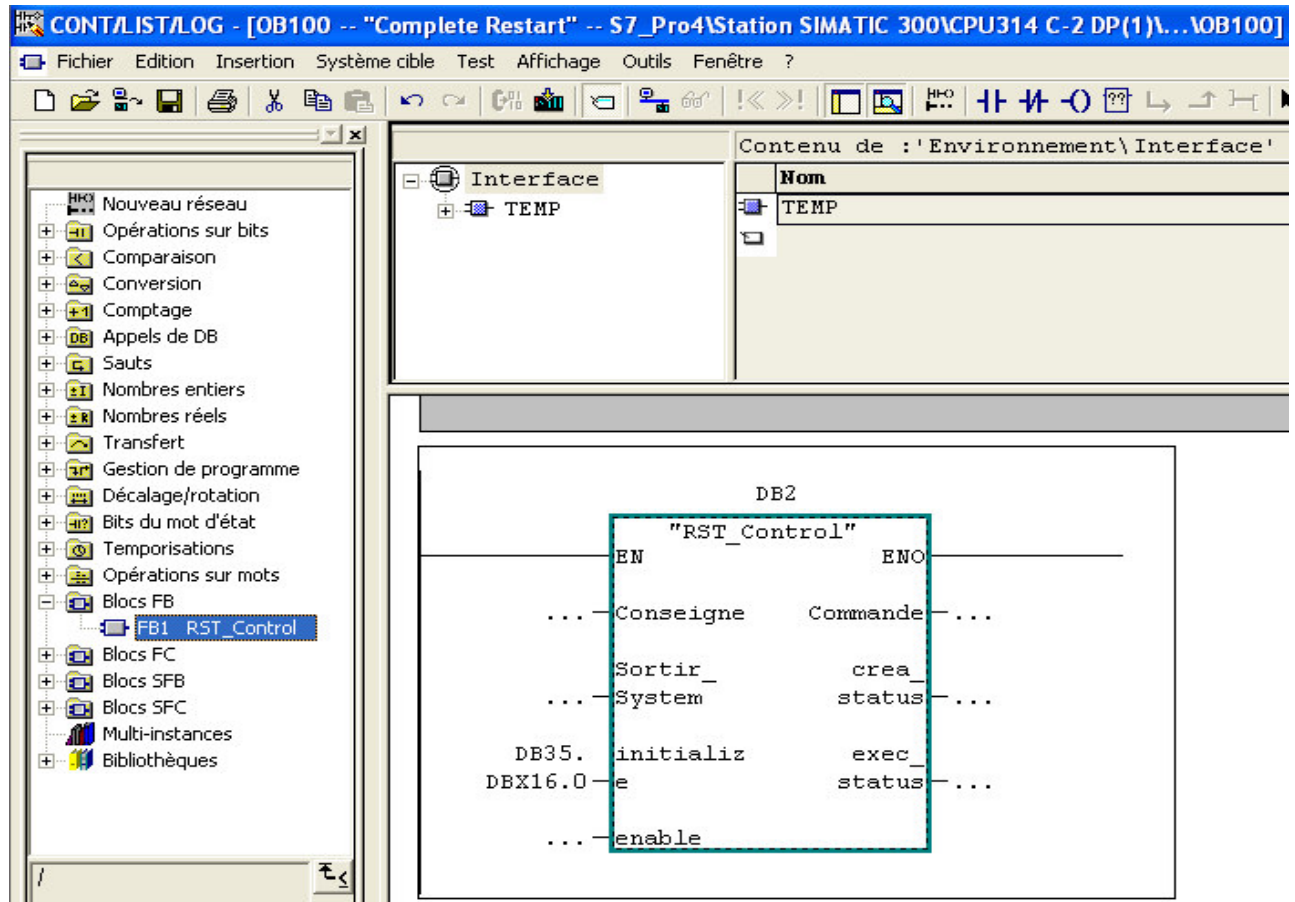


Figure III.8 : Création de block fonctionnel RST_Control

III.5 Conclusion

Nous avons vu dans ce chapitre comment avec WinAC Cible et WinAC ODK converti un modèle Simulink en code C/C++ et une source SCL et une DLL/RTDLL, et les intégrés dans programme Step7 sans savoir programmer en langage C.

Conclusion générale

Conclusion générale

Dans ce présent travail, notre objectif est l'utilisation de nouvelle stratégie d'implémentation des régulateurs , afin d'améliorer le control des système en exploitant les avantages de la commande numérique .

Dans la première partie nous avons élaborés un modèle mathématique d'un four électrique, ce modèle a été mis œuvre, en exploitant des données expérimentales extraites d'une expérience faite au laboratoire d'automates programmables et de la régulation industrielle, et en appliquant une méthode d'identification graphique, Basant sur la méthode empirique de Broïda.

Après avoir le modèle mathématique du four électrique nous avons procédés dans la deuxième partie à la synthèse d'un régulateur numérique RST afin d'exploiter les avantage de cette structure.

Et dans la dernier partie avec Matlab /Simulink de MathWorks et WinAC Cible de Siemens et via l'intermédiaire de logiciel WinAC Open Developement Kit (ODK) l'implémentation de régulateur RST devient alors possible.

Ce projet nous a permis de mettre en pratique des connaissances théoriques déjà acquises,

REFERENCES BIBLIOGRAPHIQUE

- [1] [D5910]- “four électrique à résistances. Présentation générale ”, par Jean-François BOURGEOIS, Alain GIRAULT, Richard JAUME, Marianne LE BOULCH EDF Division Recherche & Développement et Claude OBERLIN Ingénieur Senior SEE© Techniques de l’ingénieur Dossier : D5910 et Date de parution : 02/2005.
- [2] : GUERMAH Said thèse de magister, Commande adaptative d’un aérotherme par calculateur, 1998.
- [3] : Pierre Borne, Jean-Pierre Richard, Modélisation et identification des processus, Editions TECHNIP, 1992 - 369 pages.
- [4] : Par Roland Longchamp, Commande numérique de systèmes dynamiques : Cours d'automatique, Volume 2, PPUR Presses polytechniques, 2010.
- [5] : Mekki Ksouri, Pierre Borne, La commande par calculateur: application aux procédés industriels, Editions TECHNIP, 1 janv. 1999.
- [6] : Pierre Borne Jean-Pierre Richard, Régulation numérique, Editions TECHNIP, 1993.
- [7] Henri Bourlès, Linear Systems ISTE Ltd 2010
- [8] Site internet.
<http://support.automation.siemens.com/WW/view/en/569694174>

Annexe

Logiciels requise pour l'application

- Windows XP Professionnel ou Windows 7 32-bit
- STEP 7 V5.5 avec SCL V5.3 ou STEP 7 V12 SP1 ou une version supérieure
- WinAC ODK V4.2
- MATLAB / Simulink, version R2012b et plus
- WinAC Target V1.4.2

Résumé

Dans ce présent travail, notre objectif est l'utilisation de nouvelle stratégie d'implémentation des régulateurs, afin d'améliorer le control des systèmes en exploitant les avantages de la commande numérique.

Dans la première partie nous avons élaborés un modèle mathématique d'un four électrique, ce modèle a été mis œuvre, en exploitant des données expérimentales extraites d'une expérience faite au laboratoire d'automates programmables et de la régulation industrielle, et en appliquant une méthode d'identification graphique, Basant sur la méthode empirique de Broïda.

Après avoir le modèle mathématique du four électrique nous avons procédés dans la deuxième partie à la synthèse d'un régulateur numérique RST afin d'exploiter les avantage de cette structure.

Et dans la dernier partie avec Matlab /Simulink de MathWorks et WinAC Cible de Siemens et via l'intermédiaire de logiciel WinAC Open Development Kit (ODK, nous avons intégrés ce régulateur da step7.

Mots clés

Elaboration du Modèle Dynamique d'un four Électrique

Application de la Commande RST pour la Régulation de Température

Implémentation de la Commande RST sur un API S7-300