

Système chauffant control PID industriel et open source avec methode autotuning et logique floue

Arnaud SIVERT¹, Sami LABDAI¹

¹ I.U.T de l'Aisne Département Génie Electrique SOISSONS FRANCE

1 - Introduction

La régulation d'un processus est un enjeu crucial au niveau industriel pour gagner du temps dans la fabrication, pour maintenir ou suivre une consigne désirée, faire des économies d'énergies, améliorer la qualité et la sécurité, prolonger la durée de vie des machines.... Donc, la régulation permet de renforcer la compétitivité globale d'une entreprise.

D'ailleurs dans de nombreux cursus comme les départements de Génie Electrique, il y a des options automatismes industriels avec des automates et des modules spécialisés dans la régulation, mais aussi pour les départements en mécanique et génie des procédés.

Les systèmes de chauffage sont de très bons systèmes pédagogiques mais qui ont des constantes de temps un peu longues pour faire des essais pratiques en régulation. Mais pendant les tests pratiques, la modélisation et les simulations peuvent être effectuées pour trouver l'optimisation des coefficients du correcteur du régulateur.

De plus, le paramétrage d'un régulateur industriel via son manuel et la maintenance de ces systèmes avec des commandes 4-20mA peuvent être effectués pour savoir comment réagit le système en cas de dysfonctionnement et pour mettre en place des sécurités et des alarmes.

Les systèmes de chauffage sont souvent régulés pour obtenir une température constante malgré les perturbations qui peuvent être nombreuses (variation de la température ambiante, pièces incorporés dans le système chauffant, variation des paramètres en fonction de la température...)

Pour faire un régulateur sur un système de chauffage électrique, plusieurs solutions sont possibles :

- Régulation par PID (**Proportionnel, Intégral, Dérivé**) qui commande le déphasage d'un gradateur mais qui va provoquer des perturbations de courant.
- Régulation avec PWM (**Pulse-width modulation**) avec une période de hachage fixe très inférieure à la constante de temps du procédé (10 à 20 fois moins) avec un régulateur PID.
- Régulation par hystérésis (**ON-OFF hystérésis ou bang bang**) où la période des commutations est variable [1] mais où l'amplitude de variation de la sortie sera constante.

La PWM va entrainer une fréquence de commutation de l'organe de commande plus grande que celle de l'hystérésis. Mais les relais statiques **SSR** ne sont pas endommagés par ce nombre de commutation alors qu'un contacteur le sera. Etant donné que le zéro crossing de tension est fortement recommandé pour les systèmes chauffant pour éviter les contraintes CEM (compatibilité électromagnétique), le contacteur n'est en général plus utilisé. La littérature sur la régulation et les méthodes ne manquent pas surtout pour les systèmes linéaires. Mais, tous les systèmes ont des saturations et ne sont donc pas linéaires. Or, mal choisir ces coefficients de régulation peut entrainer de grands dépassements, voire pire une instabilité de la sortie, voire un temps trop long pour atteindre la consigne.

L'objectif de cet article est de présenter un système chauffant avec leurs exploitations pédagogiques mettant en œuvre différents types de régulation PID, avec une température maximale atteignant seulement 75°C par sécurité, avec une précision inférieure à 0.1°C. Un ventilateur permettra de générer une perturbation en modifiant la résistance thermique, donc la constante thermique. La réalisation devra être identique à celle d'un processus industriel de forte puissance utilisant la tension secteur mais, pour éviter un encombrement important, la partie opérative devra être réduite.

Ces réalisations utiliseront des régulateurs industriels mais aussi un programme open source. L'exploitation pédagogique pourra se faire dans le cadre de SAE (système apprentissage et d'évaluation), de travaux pratiques

d'automatique, d'automatisme, mais aussi dans le module de maintenance ou en électronique en programmant en C un microcontrôleur.

Cet article va répondre aux nombreuses questions suivantes :

- Quelle est la philosophie des régulateurs industriels ? Comment fonctionnent-ils ?
- Existe-t-il plusieurs formes de PID ? Quels sont leurs codes antiwindup et d'autotuning ?
- Existe-t-il plusieurs méthodes d'autotuning ?
- Comment remplacer un régulateur propriétaire par un régulateur open source ?
- Quels sont les outils permettant de simuler le process et le code d'un microcontrôleur ?
- Quels sont les outils permettant de confirmer la modélisation du processus et son optimisation ?
- Quels sont les appareils qui permettent de tester le 4-20mA ?

Mais dans un premier temps, nous allons présenter le système chauffant et sa modélisation.

2 - Modélisation du système chauffant et choix de commande

Le système chauffant peut-être une résistance tube de 70W de 700Ω ou un radiateur électrique de 500W [1], ou encore une lampe halogène de 70W. Cette lampe permet d'observer la puissance en fonction de l'éclairement.

A partir de l'essai à un échelon de puissance de 70W, le modèle thermique peut être effectué. Les systèmes chauffant peuvent être mis sous la forme d'un premier ordre avec retard donc une identification par Broida.

La dynamique de la température de la lampe halogène correspond à l'équation suivante :

$$Temp(t) = Power(W) \cdot RTH\left(\frac{^{\circ}C}{W}\right) \cdot \left(1 - e^{\frac{-t}{RTH \cdot CTH}}\right)$$

Avec la résistance thermique RTH déterminée en régime établi de température ayant une valeur de 0.73°C/W et la constante de temps déterminée à 63% du régime établi donnant 300s donc une capacité thermique CTH de 411J/°C.

Le modèle thermique simplifié peut être simulé dans un soft électronique avec une source de courant comme puissance comme sur la figure suivante.

L'animation de la simulation est en temps réel dans Proteus donc le temps de la simulation est aussi long qu'en pratique. Mais le tracé dans l'analog analysis se fait en quelques secondes.

Evidemment dans Scilab, il est possible de modéliser le process avec Lapalce et d'avoir la réponse du système en quelques secondes aussi.

La fonction de transfert de Laplace (s) de la température correspond à l'équation suivante

$$\frac{Temp(s)}{Power(s)} = \frac{RTH e^{-Ts}}{s\tau + 1}$$

Avec $RTH(^{\circ}C/W)$: gain statique du procédé, $T(s)$: retard (dead time), $\tau (s)$: constante de temps $RTH \cdot CTH$.

Le temps de retards est dû à plusieurs facteur (temps de réponse du capteur, emplacement du capteur, filtrage du bruit de la mesure...)

Maintenant, que le système chauffant est présenté comment peut-il être contrôlé ?

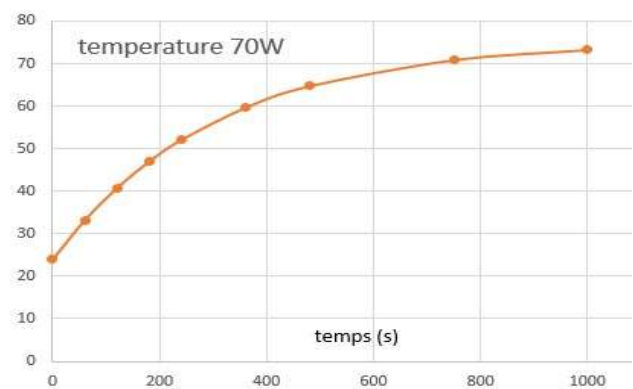


fig 1. dynamique de la température réponse indicielle pour une puissance de 70W avec une température ambiante de 20°C

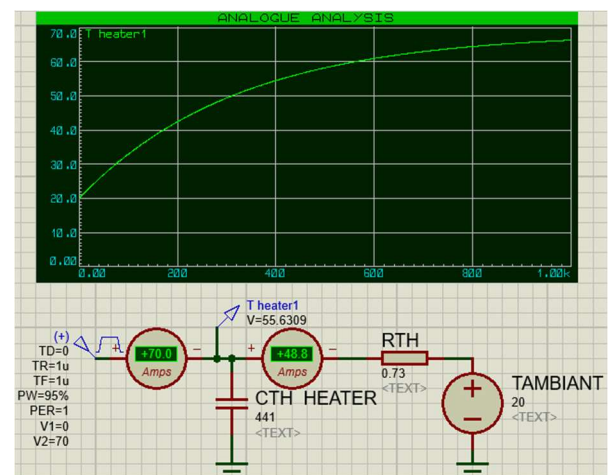


fig 2. Simulation du modèle thermique sous Proteus

3 - Régulateur PID industriel

Tous les fabricants d'automatisme (Schneider, Siemens, ABB, OMRON...) vendent des régulateurs PID autonomes.

Ces régulateurs sont numériques depuis les années 95 et ils sont devenus de plus en plus flexibles avec le temps. Mais cette flexibilité a compliqué leurs paramétrages et leurs choix d'achat en multipliant les options sur le type de capteur, de sortie (0-10V) ou boucle de courant (4-20mA), de communications, de filtrage, de régulation (PID, logique floue, neuronale...).

Etant donné que les systèmes chauffants ont des inerties importantes, le temps de calcul des microcontrôleurs ne pose pas de problèmes.

Tous ces régulateurs utilisent des afficheurs 7 segments très robustes pour donner la température de mesure en haut de l'afficheur PV (Process value) et La consigne en bas SV (Set point value) ainsi qu'un bandgraph led pour indiquer l'état de la sortie.

Tous ces régulateurs ont la même philosophie, avec des modes de communications numériques, mais souvent dédiés à leurs automates, leur IHM ou leur supervision. Pour observer les dynamiques, certains fabricants proposent aussi des softs, ou sinon il est possible de rajouter des dataloggers pour pouvoir faire l'identification comme on peut l'observer sur la figure suivante

TDE Instruments Digalox DPM72-PP Appareil de mesure numérique à encastrer



fig 3.Datalogger industriel paramétrable

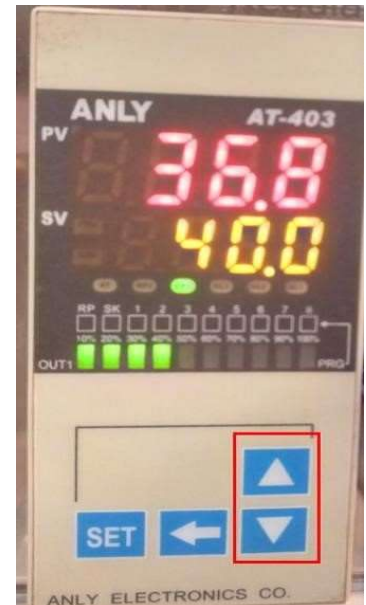
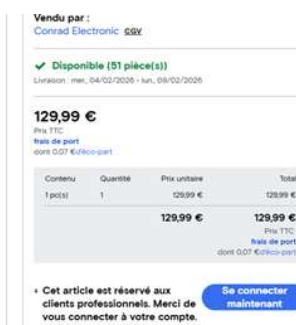


fig 4.Exemple de régulateur industriel

Voici quelques liens de manuels de différents fabricants de régulateur industriel.

https://download.schneider-electric.com/files?p_Doc_Ref=DIA3ED2090110FR&p_enDocType=Catalog
https://download.schneider-electric.com/files?p_enDocType=User+guide&p_File_Name=EIO0000001299.00.pdf&p_Doc_Ref=EIO0000001299
https://files.omron.eu/downloads/latest/manual/en/h185_e5_c-t_users_manual_en.pdf?v=1
<https://www.anly.com.tw/en/productList.php?cid=16>
https://www.tme.com/Document/03cc7f6a7ff4d65e4d866e36883a3767/at-03_datasheet.pdf
<https://www.tme.com/Document/75378d2940a4aa664b0d12f0e895fcbd/AT03-ins-en.pdf>

Ces manuels de régulateurs industriels ne donnent pas d'exemples et n'indiquent comment sont paramétrés les PID.

En effet, certains régulateurs industriels utilisent les coefficients PID dit parallèle, d'autres des PID série avec les coefficients proportionnels kps ou kp, d'autres le « proportionnel band », comme on peut l'observer sur les équations de Laplace suivantes avec ε l'erreur [7].

$$\text{Out}(s) = \varepsilon(s) \cdot kps \cdot \left(1 + \frac{1}{s \cdot Ti} + \frac{s \cdot Td}{s}\right) = \varepsilon(s) \cdot \left(kp + \frac{ki}{s} + \frac{s \cdot kd}{s}\right) = \varepsilon(s) \cdot \frac{100}{PB} \cdot \left(1 + \frac{1}{s \cdot Ti} + \frac{s \cdot Td}{s}\right)$$

Le proportionnel band est souvent utilisé car la sortie du régulateur est obligatoirement limitée. Mais les valeurs de sorties peuvent avoir différents offsets avec des sorties allant de 0 et 100%, mais d'autres variant entre -100% et 100%. Mais tout dépend comment la commande va gérer la sortie du régulateur.

De plus, la saturation de l'intégration de l'erreur surnommé anti-windup peut se faire de plusieurs façons dans la programmation des régulateurs industriels.

4 - 4-20mA, capteur, transmetteur et commande

Souvent la sortie du régulateur est une boucle de courant analogique le 4–20mA, où 4mA représente le minimum d'échelle et 20mA représente le maximum d'échelle.

Les capteurs de température PT100 et thermocouples ont besoin d'amplificateur et de pont Wheatstone car leur sensibilité est faible. Les capteurs sont souvent relativement éloignés du régulateur, donc des « transmetteurs 4-20mA » amplifient et envoient les variations de températures aux régulateurs comme on peut l'observer dans le manuel Sitranst suivant.

https://cache.industry.siemens.com/dl/files/157/109765157/att_1311958/v1/sitranst_th100_fi01_en.pdf

Si la boucle de courant est nulle, la commande et le régulateur savent qu'il y a un dysfonctionnement.

De nos jours, les boucles de courant tendent à être remplacés par le protocole **HART** (« Highway Addressable Remote Transducer », voire remplacées par des **bus de terrain numériques**. Néanmoins, une boucle de courant est plus facile à diagnostiquer, dans la mesure où un simple **multimètre** permet de mesurer la défaillance.

Pour alimenter une résistante chauffante, il y a 2 possibilités d'utilisation de relais statique utilisant des triacs :

- Soit d'utiliser un control d'angle de phase en monophasé ou triphasé comme sur la figure suivante

Type RM1E



- Relais statique CA
- Commutation analogique (contrôle d'angle de phase) pour des applications de charges résistives et de charges légèrement inductives
- Courant de commande: 4 à 20 mA ou 0 à 10 V
- Courant nominal de fonctionnement: 25, 50, 100 et 125 AC Aeff
- Tension de fonctionnement: jusqu' à 600 VCAeff
- LED de signalisation à luminosité variable en fonction du courant d'entrée
- Connecteur détrompable à verrou pour une connexion sécurisée sur versions à contrôle par tension

fig 5. Exemple de Relais statique monophasé 4-20mA avec commande à angle de phase

https://www.gavazziautomation.com/fileadmin/images/PIM/DATASHEET/FRA/SSR_RM1E.pdf

- Soit d'utiliser une commande zéro crossing en monophasé ou triphasé comme sur la figure suivante.

RZ3A



Relais statiques triphasés iphasé industriel, Commutation zéro de tension



Principales caractéristiques

- Commutation sur zéro de tension relais statique triphasé
- Valeurs nominales jusqu'à 690 VAC, 75 AAC
- Tension de blocage élevée
- Tension de commande DC ou AC
- Protection intégrée contre les surtensions
- Protection contre le dépassement de température avec sortie alarme
- LED de signalisation de l'entrée de commande et de l'état de l'alarme de dépassement de température
- Valeurs nominales des moteurs certifiées UL

fig 6. Exemple de Relais statique tout ou rien triphasé commande en tension ou courant

<https://www.gavazziautomation.com/fileadmin/images/PIM/DATASHEET/FRA/RZ3A.pdf>

Les 2 commandes PWM ou à angle de phase n'ont pas les mêmes équations :

La tension efficace avec la PWM sera linéaire.

Mais la température aura une légère fluctuation dépendant de la période et de la capacité thermique. La variable PWM correspond au rapport cyclique dans les équations suivantes. A période fixe, la variation de la température sera la plus défavorable pour un rapport cyclique de 0.5. Voici sa valeur avec un choix de période de 1s pour notre process.

$VRMS_{Secteur} := 240$ $Periode := 1$ $PWM := 0, 0.1 \dots 1$

$VRMS(PWM) := PWM \cdot VRMS_{Secteur}$

$\Delta Temperature(PWM) := \frac{power \cdot PWM \cdot (1 - PWM) \cdot Periode}{CTH}$

$\Delta Temperature(0.5) = 0.05$

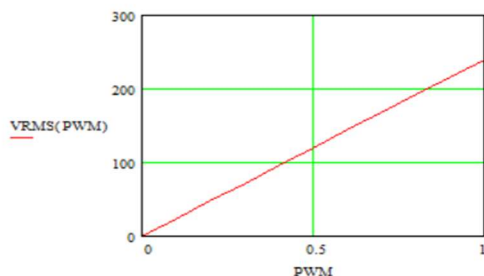


fig 7.Equation modélisation PWM

La tension efficace par angle de phase aux bornes de la résistance chauffante n'est pas linéaire comme on peut l'observer sur figure suivante.

La fluctuation de la température sera à la période secteur de 20ms.

$V_{effSecteur} := 240$ $R_{chauffage} := 700$ $\theta := 0, 0.1 \dots \pi$

$$VTRMS(\theta) := V_{effSecteur} \cdot \sqrt{1 - \frac{\theta}{\pi} + \frac{\sin(2 \cdot \theta)}{2 \cdot \pi}} \quad power := \frac{VRMS^2}{R_{chauffage}}$$

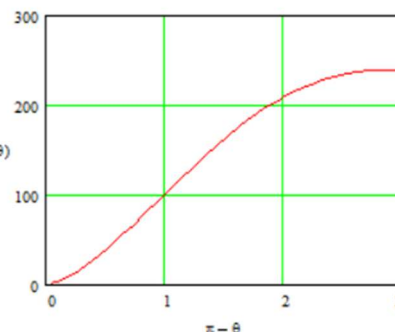


fig 8.Equation modélisation gradateur par angle de phase

Le schéma fonctionnel de la régulation du système de chauffage est représenté sur figure suivante

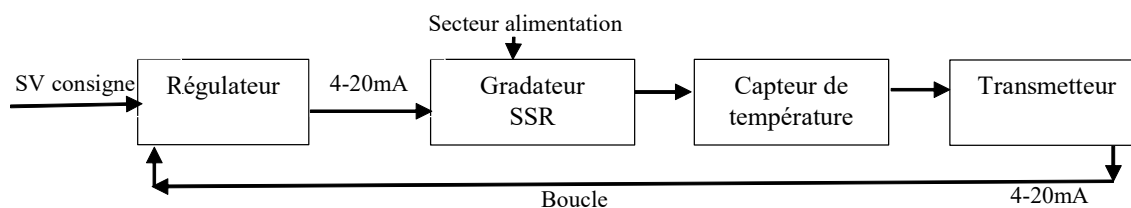


fig 9.Schéma fonctionnel d'un système chauffant électrique

Voici un exemple schéma électrique pour le régulateur chauffant avec liaison 4-20mA par angle de phase.

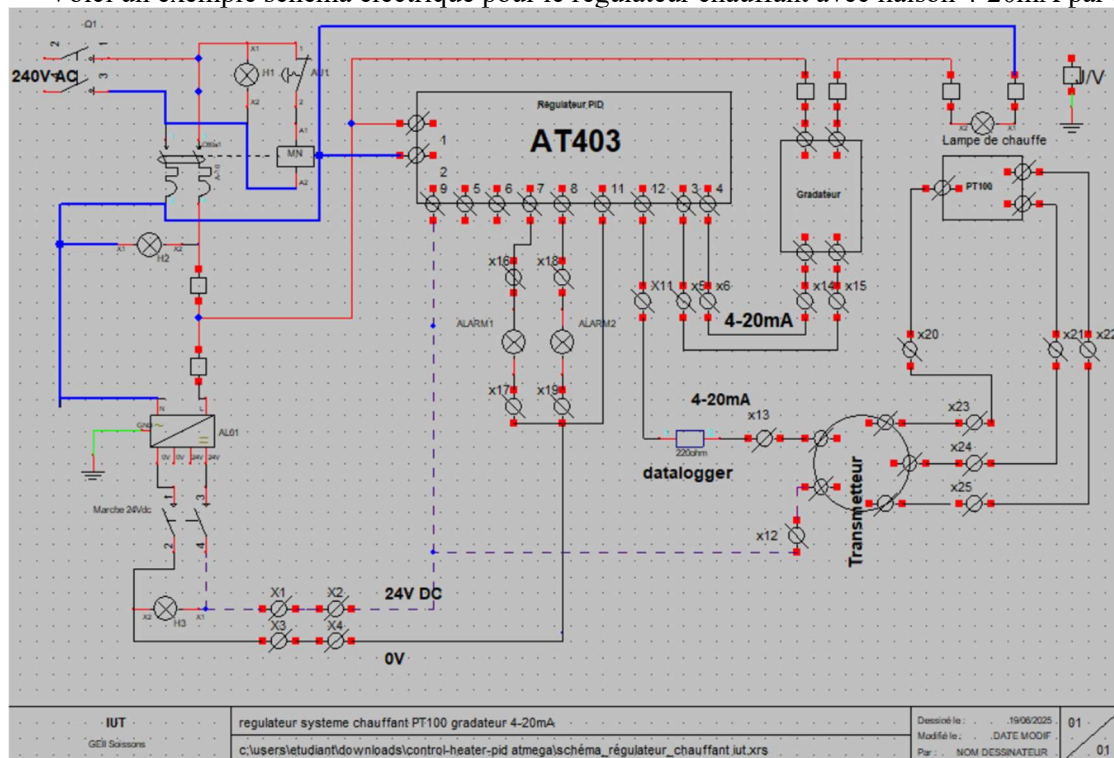


fig 10.Schéma électrique d'un système chauffant à résistance avec régulateur PID industriel

Pour gagner du temps à déterminer les coefficients d'un régulateur PID, la modélisation de la régulation du système chauffant peut-être effectuée pour vérifier son optimisation.

5 - Modélisation et simulation du système chauffant avec PID

Les systèmes non linéaires sont nombreux avec des retards de mesures, des saturations, des zones mortes, des exposants, du bruit donc chaque système aura ses propres particularités...

Les processus chauffants doivent recevoir une certaine puissance pour monter en température, définie en fonction de la capacité thermique.

Le surdimensionnement en puissance coute cher donc par pragmatisme, cette puissance ne dépasse pas 2 à 3 fois la température en régime établi désirée maximale.

D'ailleurs, le temps pour atteindre une certaine température, à partir de la température ambiante, avec une certaine puissance correspondra à l'équation suivante.

Temperature saturation entre 20°C et 50°C

$$\begin{aligned} R_{chauffage} &:= 700 & Power &:= \frac{240^2}{R_{chauffage}} & Power &= 70 & Tempsat &:= 50 \\ A &:= RTH \cdot Power & A &= 60.069 & \tau &:= RTH \cdot CTH & Tamb &:= 20 \\ Tempsat - Tamb &= A \cdot \left(1 - e^{-\frac{tf}{\tau}}\right) & tf &:= \tau \cdot \ln \left[\frac{A}{(A - Tempsat + Tamb)} \right] & tf &= 207.623 \end{aligned}$$

fig 11. Equation du temps pour atteindre une certaine température (50°C ici) pour une puissance constante.

Avec la saturation de la commande, seul la méthode SIMC (**Simple Internal Model Control**) [2, 5] permet de déterminer les coefficients du PID avec une zone linéaire importante autour de la consigne.

Dans un premier temps, la modélisation de la régulation avec PWM va être présentée.

5.1 - Régulation PID avec PWM

Par simplification, le modèle de la commutation de la PWM ne sera pas pris en compte. De même, étant donné que la période d'échantillonnage est très faible par rapport à la constante de temps du processus, la régulation sera sous forme de fonction de transfert de Laplace et non sous forme discrétisée.

On utilisera le PID parallèle pour que les étudiants puissent observer l'action de chaque coefficient. Etant donné que la sortie out est une PWM, celle-ci variera de 0 à 100%, donc avec une saturation de sa sortie. De plus, l'intégration de l'erreur sera aussi saturée appelé communément anti-windup pour que l'intégration ne se fasse pas pendant la saturation de la commande. De plus, par simplification, le coefficient dérivé ne sera pas utilisé.

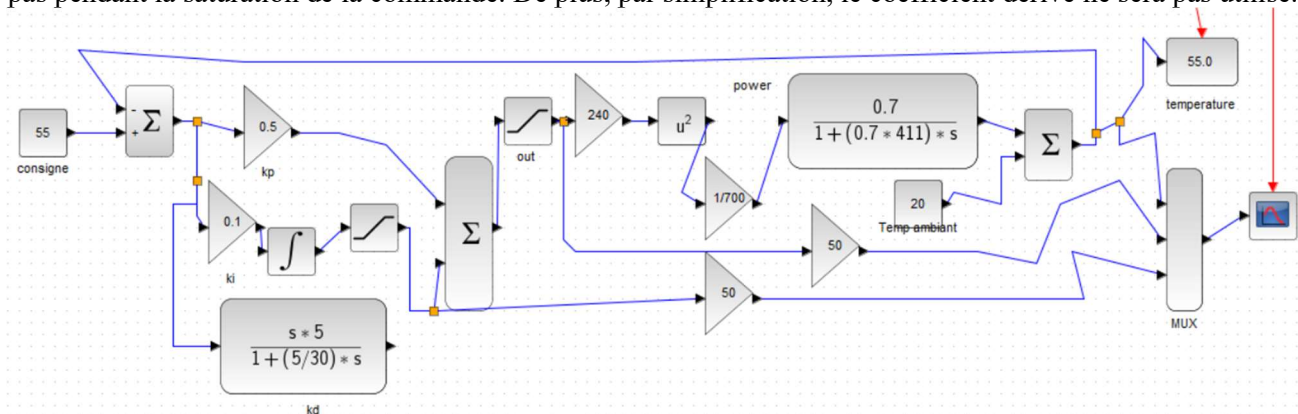


fig 12. Modèle Scilab d'un système chauffant à PWM et période d'échantillonnage de période de 1s

A partir de la modélisation en boucle ouverte, il est possible de déterminer la fonction de transfert de la boucle fermée pour savoir comment agissent les coefficients du PID. Grâce au coefficient intégral, la température en régime finale correspondra à la consigne malgré les perturbations. La preuve peut être faite en faisant, la limite pour $t \rightarrow \infty$, de la fonction de transfert de la boucle fermée de l'équation de la figure 13.

Un filtre numérique pour éliminer le bruit de la mesure de la température provoque un retard de la mesure de 5s qui pourra être négligé par rapport à la constante de temps du process qui est de 300s. Hors saturation de la sortie out, avec un régulateur PI, la fonction de transfert du système bouclé correspondra à un système du deuxième ordre avec z correspond au coefficient d'amortissement et ω_0 à la pulsation du système déterminant la constante de temps dominante.

Si le coefficient d'amortissement z est supérieur à 1 alors le système est apériodique (sans dépassement). Le correcteur PI permet de compenser la constante de temps dominante et d'imposer la constante de temps τ_2 qui a été simplifiée.

A partir des équations, plus le coefficient k_p sera grand et plus la dynamique en boucle fermée sera rapide. Plus k_i sera grand et plus coefficient d'amortissement diminue donc il y a risque de dépassement.

D'ailleurs, la méthode SIMC détermine les coefficients du PID à partir du modèle en boucle ouverte du premier ordre avec retard avec les équations de la figure 14.

Avec $k_c=5$, la zone linéaire, sans saturation autour de la consigne, correspondra à l'équation suivante :

$$T = \frac{out_{max}}{kp(\%/^{\circ}C)} = \frac{100\%}{kp} = \frac{100}{8} = 12.5^{\circ}C$$

Cette méthode compense la constante de temps dominante lente τ_d qui tend à correspondre à la constante de temps en boucle fermée et va imposer sa constante de temps τ_c qui correspondra à la fonction de la figure 15.

Par conséquent, plus le coefficient k_c est grand et plus le système en boucle fermée sera rapide mais plus la zone de non saturation autour de la consigne va se restreindre. D'ailleurs, la réponse de la température sans saturation avec $k_c=5$ correspond à la figure 16.

Evidemment, la simulation de la boucle fermée sous Scilab permet de vérifier les équations mathématiques et d'observer la saturation de la sortie du régulateur comme on peut l'observer sur la figure 18 avec une température ambiante de $20^{\circ}C$.

$$\begin{aligned} k_p &:= 0.078 & k_i &:= 0.00024 & T &:= 5 \\ consigne &:= 55 & RTH &:= 0.7 & CTH &:= 411 & \tau &:= RTH \cdot CTH & Tamb &:= 50 \\ U &:= 240 & R &:= 700 \\ A &:= \frac{U^2 \cdot RTH}{R} & A &:= 57.6 \\ z &:= \frac{1 + k_p \cdot A}{2 \cdot \sqrt{k_i \cdot \tau \cdot A}} & z &:= 1.373 \\ \omega_0 &:= \sqrt{\frac{k_i \cdot A}{\tau}} & \frac{1}{\omega_0} &:= 144.262 \\ csttempsdominante &:= \frac{1}{\omega_0 \cdot (z - \sqrt{z^2 - 1})} & csttempsdominante &:= 333.948 & \frac{k_p}{k_i} &:= 323.93 \\ Temp(s) &:= \frac{consigne \cdot \left(kp + \frac{k_i}{s} \right) \cdot \left(\frac{A \cdot e^{-T \cdot s}}{1 + \tau \cdot s} \right) + Tamb}{1 + \left(kp + \frac{k_i}{s} \right) \cdot \left(\frac{A \cdot e^{-T \cdot s}}{1 + \tau \cdot s} \right)} \\ Temp(s) &:= \frac{consigne \cdot \left(\frac{kp}{s} + 1 \right) + \frac{Tamb}{k_i \cdot A} \cdot (1 + \tau \cdot s) \cdot s}{1 + s \cdot \frac{1 + kp \cdot A}{k_i \cdot A} + s^2 \cdot \frac{\tau}{k_i \cdot A}} \\ TempA(s) &:= \frac{consigne + \frac{Tamb}{k_i \cdot A} \cdot s}{(1 + \tau_2 \cdot s)} & \tau_2 &:= \frac{1}{\omega_0 \cdot (z + \sqrt{z^2 - 1})} & \tau_2 &:= \frac{\tau}{kp \cdot A} & \tau_2 &:= 62.32 \end{aligned}$$

fig 13. Fonction de transfert en boucle fermée PWM du système chauffant

$$\begin{aligned} k_c &:= 5 & \tau_c &:= \frac{\tau}{k_c} & k_p &:= \frac{\tau}{A \cdot (\tau_c + T)} & k_i &:= \frac{k_p}{\tau} & k_d &:= kp \cdot \left(\frac{\tau \cdot T}{2 \cdot \tau + T} \right) \\ k_p &:= 0.08 & k_i &:= 2.776 \cdot 10^{-4} & k_d &:= 0.198 \\ wc &:= \sqrt{\frac{1}{\tau \cdot (\tau_c + T)}} & \frac{1}{wc} &:= 134.137 \\ zc &:= \frac{1}{2} \cdot \sqrt{\frac{\tau}{\tau_c + T}} & zc &:= 1.072 \\ \tau_d &:= \frac{1}{wc \cdot (zc - \sqrt{zc^2 - 1})} & \tau_d &:= \frac{2 \cdot \tau}{(\sqrt{kc} - \sqrt{kc - 4}) \cdot \sqrt{kc}} & \tau_2 &:= \frac{2 \cdot \tau}{(\sqrt{kc} + \sqrt{kc - 4}) \cdot \sqrt{kc}} & \tau_2 &:= 79.518 \\ Temp2(s) &:= \frac{(consigne + Tamb \cdot (\tau_c + T) \cdot s) \cdot (\tau \cdot s + 1)}{(1 + s \cdot \tau_d) \cdot (1 + s \cdot \tau_2)} \end{aligned}$$

fig 14. Détermination des coefficient PI pour avec une constante de temps τ_c méthode SIMC

$$\begin{aligned} wc &:= \sqrt{\frac{1}{\tau \cdot \tau_c}} & zc &:= \frac{1}{2} \cdot \sqrt{kc} & \tau_2 &:= \frac{\tau}{kc} & \tau_2 &:= 57.54 \\ Temp2(s) &:= \frac{consigne + Tamb \cdot (\tau_c + T) \cdot s}{(1 + s \cdot \tau_2)} \end{aligned}$$

fig 15. PID SIMC, nouveau coefficient de la fonction de transfert en Boucle fermée en négligeant le retard

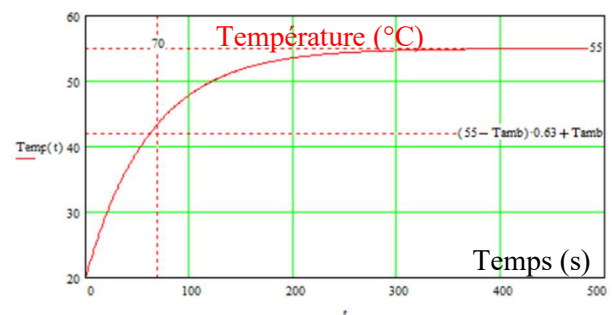


fig 16. Réponse de la fonction de transfert lorsqu'il n'y a pas de saturation de la puissance pour atteindre $55^{\circ}C$ avec les coefficient PI précédente.

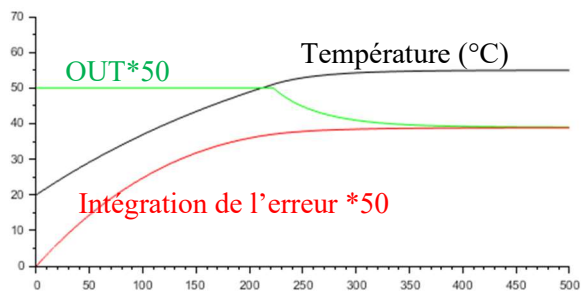


Fig 18 Réponse de la modélisation PWM en boucle fermée pour $k_p=0.08$, $k_i=0.00027$ avec température ambiante de 20°C

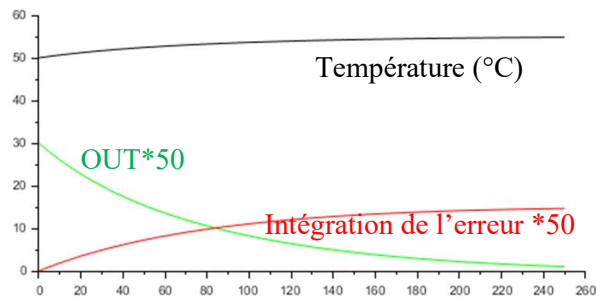


fig 17. Réponse de la modélisation PWM en boucle fermée pour $k_p=0.08$ et $k_i=0.00027$ avec température amb de 50°C

Evidemment, avec une température ambiante de 50°C (figure 17), il n'y a pas la saturation de la sortie et l'on retrouve les dynamiques de la théorie malgré les quelques simplifications les équations effectuées.

En synthèse, sur le régulateur PID parallèle, si on agit sur un des coefficients qu'est ce qui se passe sur le process ?

Réponse PID	Temps de réponse	dépassement	Erreur statique
Si k_p augmente	diminue	diminue	réduit
Si k_i augmente	Peu de changement	augmente	Elimine plus rapidement
Si k_d augmente	Pas de changement	diminue	Pas de changement

D'où la méthode empirique du « règleur » surnommé (Trial-and-error tuning) ou l'on augmente le coefficient proportionnel jusqu'à ce qu'il y un début de dépassement, puis une augmentation du coefficient intégral pour éliminer plus rapidement l'erreur statique jusqu'à la limite d'un petit dépassement, puis une augmentation du coefficient dérivée pour minimiser le dépassement.

[https://eng.libretexts.org/Bookshelves/Industrial_and_Systems_Engineering/Chemical_Process_Dynamics_and_Controls_\(Woolf\)/09%3A_Proportional-Integral-Derivative_\(PID\)_Control/9.03%3A_PID_Tuning_via_Classical_Methods](https://eng.libretexts.org/Bookshelves/Industrial_and_Systems_Engineering/Chemical_Process_Dynamics_and_Controls_(Woolf)/09%3A_Proportional-Integral-Derivative_(PID)_Control/9.03%3A_PID_Tuning_via_Classical_Methods)

Comme pour la régulation avec PWM, la régulation par angle de phase peut être aussi facilement simulable.

5.2 - Modélisation et Régulation à angle de phase

A partir des équations du gradateur monophasé par angle de phase, le modélisable peut être mis sous la forme de la formule suivante avec un régulateur PI.

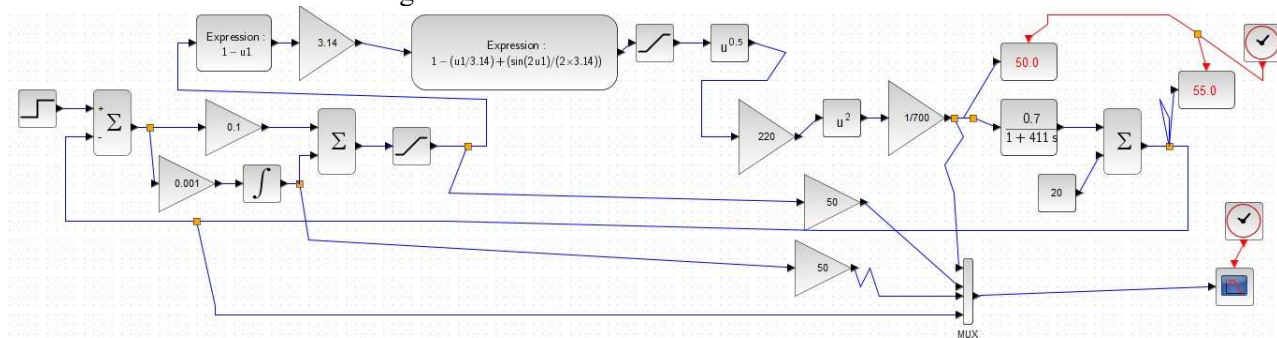


fig 19. Modèle Scilab en boucle fermée d'un système chauffant à angle de phase avec sortie saturable de 0 à 100%

Cet article ne détaillera pas les réponses de ce modèle à angle de phase qui ressemble fort à celle de la PWM. Par contre, pour les étudiants en informatique embarquée, il est possible de remplacer le régulateur industriel par un microcontrôleur.

6 - Régulateur microcontrôleur PID et datalogger open source

Un régulateur PID ne demande pas beaucoup de calculs et ce n'est pas la peine de passer par des « Library » de programme où l'on ne connaît pas les tenants et les aboutissants de leurs calculs.

La réalisation de boucle 4-20mA est très facile à faire en électronique [7] comme le montre le schéma de la figure suivante. La résistance datalogger de 250Ω dans la boucle de courant permet d'avoir une image en tension 5V de la boucle 4-20mA. En effet, lors de la maintenance mesurer des courant de 20mA n'est pas si facile avec des pinces de mesure de courant à effet hall.

Par contre, si le datalogger est en 3.3V, la résistance passera à 165Ω.

Avec une alimentation de 24V, il est possible de mettre 3 datalogger et 1 commande qui auront des résistances de 250 Ω.

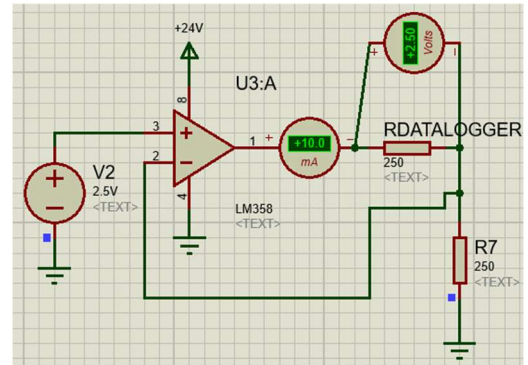


fig 20. Convertisseur 0-5V en 0-20mA utilisant un AOP.

La mesure de la température peut avoir un peu de bruit donc un filtre numérique passe bas est souvent utilisé mais entrainera un léger retard négligeable par rapport à la constante de temps du process.

Pour tester le code du microcontrôleur, une simulation du programme sous Proteus est bien pratique.

Un microcontrôleur ATMEGA 328 a été utilisé avec une routine d'interruption de 10ms pour faire une PWM avec une période de 1s pour une précision du rapport cyclique de 1%.

De multiples capteurs de température peuvent être utilisés aussi en simulation sous Proteus.

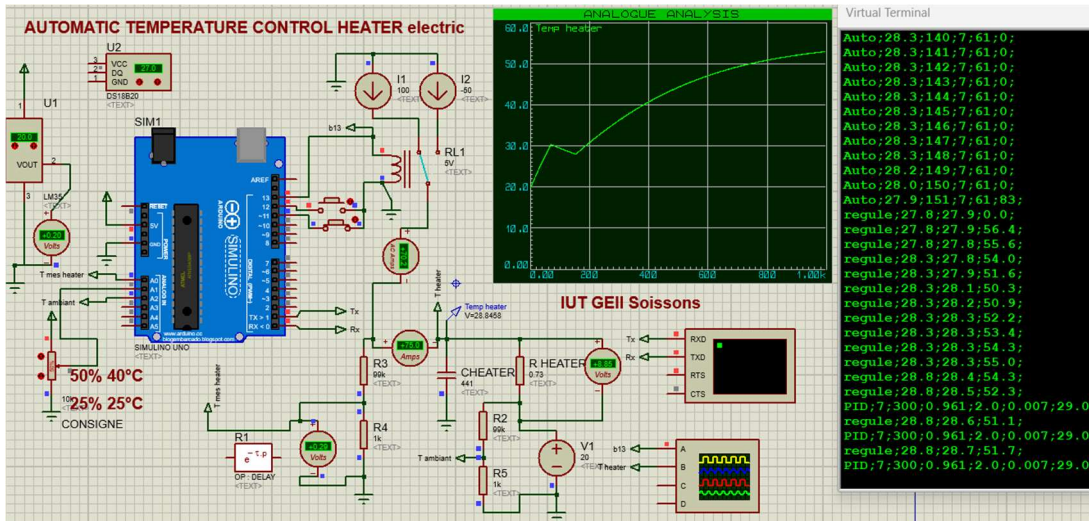


fig 21. Schéma électrothermique pour vérifier le code du la régulation du PID et de l'autotuning [3]

Le codage en C, de la PWM et du PID ne prennent que quelques lignes, donc inutile de prendre des « libraries » sous programme [6].

```
tempson=PWM ; //temps On=PWM*periode/(timerISR*100)=PWM*1s/(0.01*100)
if (compt<tempson) {digitalWrite(13,1) ;}
if (compt>tempson) {digitalWrite(13,0) ;}

erreurl=erreur;
erreur=consigne-Tfiltre;
integral=ki*erreur+integral;
derive=kd*(erreur-erreurl); //
if (integral>=90) {integral=90;} //
if (integral<=-90) {integral=-90;}

PWM=kp*erreur+integral+derive;
if (PWM>=100) {PWM=100;} // PWM en %
if (PWM<=0) {PWM=0;}

```

Les méthodes autotuning pour déterminer les paramètres du PID sont nombreuses avec des identifications par des réponses à un échelon ou fréquentielles. D'ailleurs, tous les régulateurs industriels le proposent sans expliquer la méthode qu'ils utilisent dans leurs manuels.

Mais quelle est la meilleure des méthodes et la plus rapide ? Comment la coder en C ?

7 - Autotuning PID

L'autotuning est la recherche par le microcontrôleur de l'identification du système à un moment donné, puis de déterminer les coefficients du PID.

La méthode la plus rapide pour identifier un système est celle surnommé « relay commande » qui impose une commande en tout ou rien à période fixe (signal carré) au process. Sinon mesurer les variations de la sortie lorsque la sortie du régulateur est à 100% puis lorsque la sortie est à 0%

En effet, lorsqu'il y a une commande d'un système de première ordre avec une commande en tout ou rien à période fixe (signal carré) alors la réponse de la sortie correspondra à la figure suivante.

On peut observer les incréments des pentes montantes et descendantes de la sortie en fonction de l'entrée, puis l'amplitude des oscillations en régime établi.

En régime établi, la température moyenne sera égale au gain du système multiplié par la puissance moyenne.

La période du « relay command » doit être bien plus grande que celle de la PWM et bien inférieure à la constante de temps du système. Une période de 10% de la constante de temps du système est un bon compromis.

Une autre solution est que le micro mesure le temps pour une montée de température significative (au moins 10°C) puis que le micro mesure le temps en roue libre d'une diminution de température de 2°C. D'ailleurs, sur la figure 21, on peut observer cette méthode d'identification.

Voici le codage en C de cet autotuning, on peut remarquer que l'identification et le calcul des coefficients du PID ne prend que quelques lignes

```
if (flagautotuning==1) {
    //methode relay control roue libre
    if (timeauto==0) {Tinit=Tfiltre; PWM=100; flagretard=1; }
    Tamb=analogRead(A2);
    Tamb=Tamb/2.048;
    timeauto++;
    if( (Tfiltre>(Tinit+0.5)) && flagretard==1 ) {retard=timeauto; flagretard=0; flagton=1; }
    if( (Tfiltre>=(Tinit+10)) && flagton==1 ) {ton=timeauto-retard ; flagton=0; flagtoff=1; PWM=0; }
    if( (Tfiltre<=(Tinit+10-2)) && flagtoff==1 ) {toff=timeauto-ton-retard ; flagtoff=0; flagPID=1; }

    if (flagPID==1) {
        csttemps=toff/(log(Tinit+10-Tamb)/(Tinit+10-2-Tamb)); //identification pour -2°C
        CTH= power/((10/ton)+(Tamb/csttemps)) ;
        RTH=csttemps/CTH;

        lambdaCstBF=(RTH*CTH)/2; //methode SIMC choix de la constante de temps en BF
        kp=(CTH)/(retard+lambdaCstBF);
        ki=(kp)/csttemps;
        kd=(CTH*T)/(2*lambdaCstBF); flagPID=0; flagautotuning=0; }
}
```

L'autotuning demande environ 10% du temps la constante de temps en boucle ouverte pour faire une identification pertinente sur un système de chauffage.

Ne pas confondre l'autotuning et l'adaptive tuning ou ce dernier va rechercher en permanence les paramètres du modèle et recalculer les coefficients du PID.

Après avoir paramétré le process, il faut s'interroger sur différents dysfonctionnements possibles de ce système chauffant et rajouter des alarmes et couper le chauffage en cas de défaut.

Mais que vont provoquer certains dysfonctionnements sur le process ?

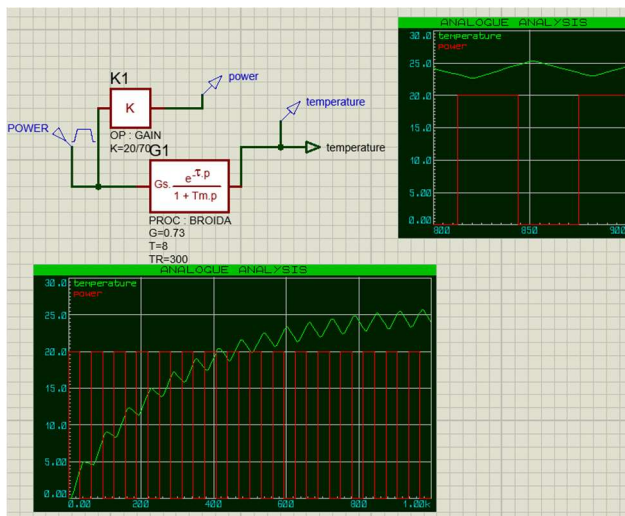


fig 22. Réponse à un signal carré d'un système du premier ordre avec retard

8 - Maintenance

En maintenance, il faut trouver par des mesures pourquoi le système ne fonctionne plus et quels sont le ou les éléments qui peuvent être incriminés lors de telle ou telle action du système. Voici quelques interrogations :

- Que se passe-t-il si le capteur de température est en défaut ? ou que le transmetteur 4-20mA du capteur est défectueux ? ou que le transmetteur n'a plus d'alimentation 24V ?
- Que se passe-t-il si l'alimentation secteur est coupée ? ou que le gradateur ne fonctionne plus

Dans les 3 cas, le régulateur industriel indique sa température minimale de -25°C et le système chauffe à 100% alors que le régulateur industriel devrait afficher une erreur de la boucle de courant

Des maquettes sur grille existent pour appréhender la maintenance de système chauffant.

Sur cette maquette un microcontrôleur ESP32 a été rajouté pour faire datalogger [3] ainsi que pour remplacer le régulateur industriel.

Les étudiants de génie électrique option électronique ont pu savoir comment programmer un PID en codant leur propre régulateur.

Les étudiants de génie électrique option automatisme ont pu se confronter à la difficulté de maîtriser le manuel de régulateur industriel.

Aujourd'hui, il est rare de trouver un régulateur industriel que l'on peut paramétrer en C++ ou via Matlab ou via une supervision. Pourtant, cela permettrait de mieux contrôler le processus et d'avoir plus de vente.

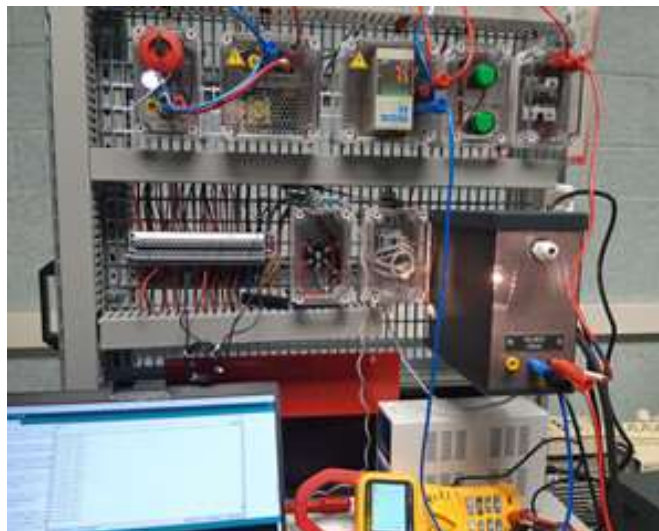


fig 23. Assemblage des différents éléments faisant un régulation de température autonome sur une grille avec bornier pour pouvoir faire la maintenance [4].

De plus, avoir des régulateurs programmables en open source, cela permettrait d'améliorer les processus et le codes pour que cela corresponde à une multitude d'utilisateurs.

Depuis 2023, pour faire la maintenance des entrées 4-20mA très abordables, des petits appareils génèrent le signal de courant et permettent de remplacer un transmetteur donc, de vérifier si l'entrée du régulateur fonctionne bien ou de vérifier sa calibration ou de tester un transmetteur. Ces générateurs sont aussi multimètres en entrée et permettent de tester nombreux types de capteurs comme on peut l'observer sur la figure suivante. Cet appareil fait aussi data logger avec écran et mémoire mais aussi sur sa sortie USB C

Des « process meter » existe depuis longtemps, tel que Fluke 789.

Certains régulateurs industriels proposent une « régulation floue » à la place de faire du PID. Mais à quoi cela correspond ?



fig 24. Générateur de signal multifonction FNIRSI® SG-004A, input and output

9 - Régulateur Logique floue

La littérature sur la logique floue n'est pas facile à comprendre avec ses règles d'appartenances et de fuzzyfication. La logique floue permet de :

- tolérer l'imprécision, d'où son terme flou
- travailler avec des **règles linguistiques** avec des systèmes mal connus et non linéaires.

Exemple de règles linguistiques pour des process de chauffage

- Si l'erreur est **grande positive** $>12.5^{\circ}\text{C}$ → chauffage **fort 100%**
- Si l'erreur est **petite positive** $<6.25^{\circ}\text{C}$ → chauffage **moyen 50%**
- Si l'erreur est **égale à 0°C** → chauffage **éteint 0%**
- Si l'erreur est **négative** $<0^{\circ}\text{C}$ → chauffage **éteint 0%**

Donc, la sortie du régulateur peut s'écrire tout simplement $\text{Out}(\%) = k1. \varepsilon$ avec $k1 = 100\%/12.5^{\circ}\text{C}$.

Même si la précision de la mesure de la température est faible avec une valeur de 0.5°C . Mais, avec le filtre passe bas numérique fait par le processeur cette imprécision passe à 0.01°C .

Pour une implémentation aisée sur un microcontrôleur, le contrôleur flou est basé sur une table utilisant l'erreur et sa dérivée, puis il y a une intégration de la sortie des valeurs de cette table pour annuler les perturbations. Avec la defuzzification, la sortie avec l'intégration du régulateur flou est linéaire et revient à un PI (proportionnel intégral d'où le nom « fuzzy logique PI ») [8]

La commande du régulateur out est directement déduite de cette table appelé plan de phase où il faut définir les facteurs d'échelles de l'erreur et de sa dérivée discrétisée ($k1. \varepsilon, k2. \Delta \varepsilon$). Dans cette table, il y a une ligne de 0 surnommée la ligne de commutation où la trajectoire du système glisse autour de cette ligne de 0 si elle est bien attirée dont l'équation de Laplace est la suivante :

$$\text{Ligne commutation} = k1. \varepsilon + k2. \varepsilon. s = 0$$

Par conséquent, la sortie du régulateur va agir comme un proportionnel intégral correspondant à l'équation de Laplace suivante

$$\text{Out} = \varepsilon. (k1 + k2. s). k3 / s = \varepsilon. (k1 / s + k2). k3 = \varepsilon. (ki / s + kp)$$

Le facteur d'échelle $k1$ influe sur la saturation de l'erreur de la zone linéaire autour de la consigne.

En effet, si $k1 = 0.1$ avec une erreur de 10°C , la sortie s'incrémentera de 1% avec un coefficient $k3$ de 1.

Les rapport $k2/k1$ déterminent la constante de temps de la sortie correspondant à l'équation discrétisée suivante s'il y a l'attractivité autour de la ligne de commutation

$$\text{Temp}(n. Te) = (T_{\text{ambiante}} - \text{Consigne}) \cdot e^{-\frac{k1}{k2} n Te} + \text{Consigne}$$

Pour avoir une constante de temps de $70s = k2/k1$ donc $k2 = 7$ pour $k1 = 0.1$

Donc, le choix des coefficients $k1, k2, k3$ n'ont rien à voir avec la méthode SIMC même si on retrouve les mêmes dynamiques.

Voici la simulation avec les choix des facteurs d'échelles précédents [3] qui permet de vérifier la programmation.

Le codage en C du fuzzy logic est le suivant

```
erreur1=erreur;
erreur=consigne-Tfiltre;
derreur=(erreur-erreur1);
fuzzy=k1*erreur+k2*(erreur-erreur1);
if (fuzzy>=100) {fuzzy=100;} //
if (fuzzy<=-100) {fuzzy=-100;}

integral=k3*fuzzy+integral;

if (integral>=100) {integral=100;} //
if (integral<0) {integral=0;}
PWM=integral;
```

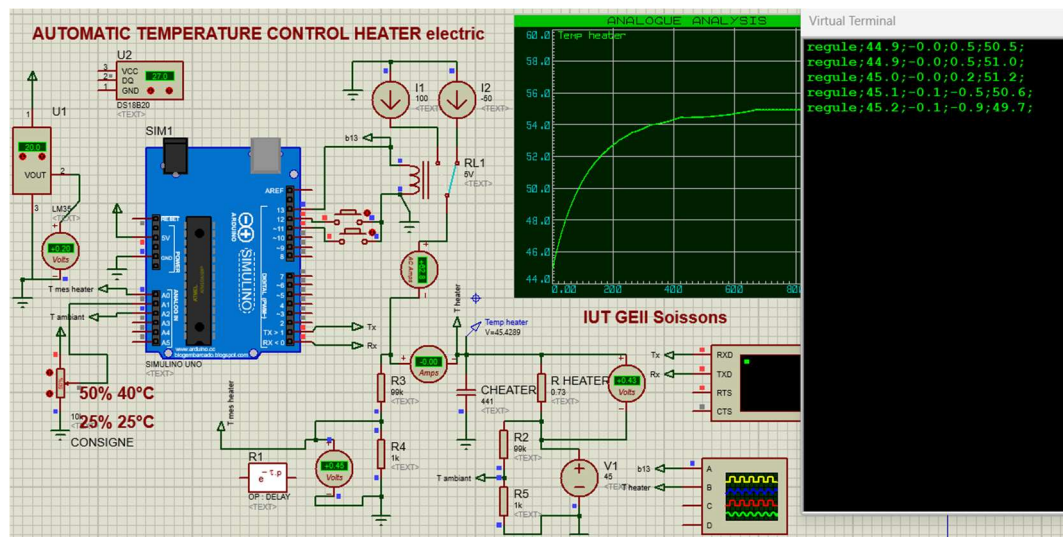


fig 25.Schéma électrothermique pour vérifier le code du la régulation par logique floue [3]

Pour vérifier les dynamiques de la logique floue, le modèle suivant a été utilisé en discrétisant le régulateur flou dans Scilab avec une période d'échantillonnage de 1s.

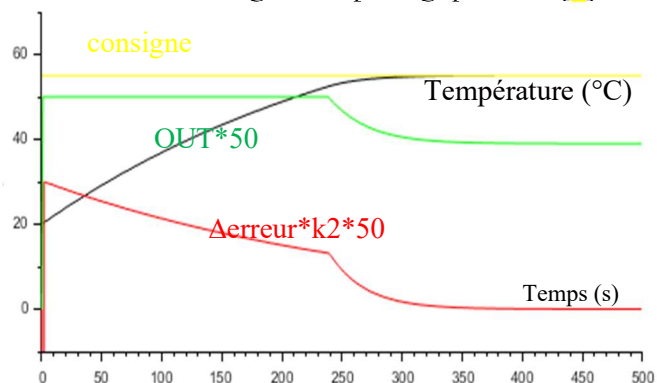


fig 26.Réponse de la régulation avec logique floue pour $k_1=0.1$, $k_2=3$, $k_3=1$

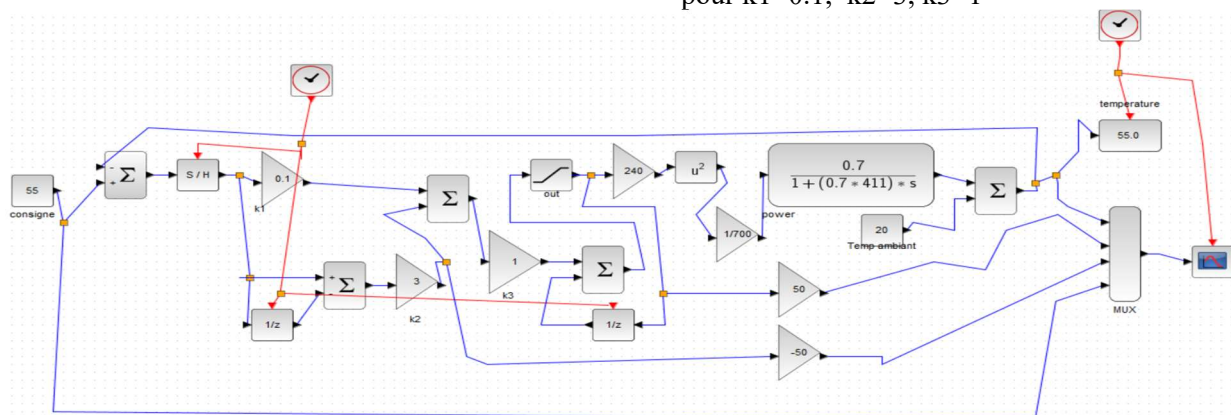


fig 27.Modèle Scilab en boucle fermée d'un système chauffant contrôler avec un régulateur flou avec sortie saturable de 0 à 100%

En synthèse sur les facteurs d'échelles de la logique floue, si on agit sur les coefficients que se passe-t-il sur le process ?

Réponse PI flou	Temps de réponse	dépassement	Erreur statique	Autour consigne
Si k_1 augmente	diminue	augmente		
Si k_2 augmente	augmente	diminue		
Si k_3 augmente			Elimine plus rapidement	Provoque variation

Comment paramétrer différentes consignes séquentiellement ?

10 - Paramétrage séquentiel de consigne (échelon, rampe...)

Les régulateurs industriels PID proposent un paramétrage séquentiel de différentes valeurs de consignes ou de rampes et certains permettent même un fonctionnement en étapes via un grafcet.

Les machines d'état avec [grafcet](#) permettent d'avoir une programmation simple et très robuste en process donc avec peu de bugs de programmation.

En effet, un automate programmable n'est qu'un micro contrôleur où il est possible de rajouter des racks spécifiques avec un multiplexage des entrées et sorties. Certains utilisent des cartes arduinos, ESP32, STM32 avec une programmation par grafcet [9]

Pourtant, la retranscription du langage grafcet (algorithme) en langage structuré C est facile à faire.

Bref, les rampes sont très faciles à programmer en C avec des fonction récurrentes.

Mais malgré le régulateur PI sur un système de premier ordre, avec une rampe, il y aura une « erreur de trainage » (tracking errors). Cette erreur sera égale à la rampe multipliée par la constante de temps dominante en boucle fermée comme démontrée sur ce lien

https://public.iutenligne.net/automatique-et-automatismes-industriels/verbeken/cours_au_mv/chapitre3/chap34.html

$$\text{Erreur suivie } (^{\circ}\text{C}) := \text{pente rampe } \left(^{\circ}\frac{\text{C}}{\text{s}}\right) \cdot \text{constante temps BF } (s)$$

D'ailleurs, il y a des méthode d'indentification pour identifier la constante de temps dominante d'une régulation en boucle fermée avec l'utilisation d'une rampe en consigne à cause de cette erreur de suivie dans la zone de la régulation linéaire car il y a évidemment la saturation du système.

Donc, il faut que la rampe soit inférieure aux temps de montée de la température, dans notre cas pour être dans cette zone linéaire.

Evidemment, cette erreur s'annulera lorsque la consigne redeviendra une constante.

Exemple pour la variation de consigne suivante

```
if (digitalRead(10)==0) { //rampe cycle process
    if (seconde==0 ) {consigne=20;}
    if (seconde<200 ) {consigne=0.1+consigne;} //rampe integration recurente discretisée jusqu'a 40°C en 200s ((40-20)/200)=0.1
    if (seconde>200 && seconde<300) {consigne=40;} //consigne constante pendant 50s
    if (seconde>=300 && seconde<400) {consigne=(0.05+0.025)+consigne;} //consigne=((45-40)/100)+consigne + erreur de suivie
    if (seconde>=400 && seconde<500) {consigne=45;}
    if (seconde>=500 && seconde<cycle) {consigne=55;}

    if (seconde>cycle) {seconde=0;} //faire un nouveau cycle
}
seconde++;
```

Sur la figure suivante, on peut observer que pour les 200 premières secondes, cette erreur de suivie pour la rampe est de 0.1°C/s. D'ailleurs, il y a 7°C d'erreur à t=200s donc que la constante de temps est déterminée par l'équation suivante :

$$\text{constante temps BF}(s) = 7^{\circ}\text{C} / 0.1^{\circ}\text{C/s} = 70s$$

Etant donné que l'on connaît cette erreur de suivie, alors il est possible de le rajouter sur la rampe pour minimiser l'erreur.

Exemple pour une pente désirée de 0.05°C/s, pour aller de 40°C à 45°C en 100s, provoquant une erreur de suivie de 3.5°C, la rampe devrait être de 0.05°C/s + 3.5°C/100s = 0.085°C/s, mais il y a un risque d'un petit dépassement. D'ailleurs, on peut observer que l'erreur de suivie entre 200 et 300s est presque nulle, ainsi que l'erreur statique pour la consigne de 45°C.

Puis, il y a la réponse à un échelon pour atteindre les 55°C.

12 - Bibliographies

- [1]. A.Sivert, A.FAQIR « Radiateur électrique open source, réparabilité, durabilité, commande, identification de l'isolation pièce »
https://sti.eduscol.education.fr/ressources_techniques/radiateur-electrique-open-source-et-ses-modeles
<https://forum.arduino.cc/t/control-heater-automatic-open-source-regulation-hysteresis-de-temperature-radiateur-reparable-simulation/>
- [2]. F.Mudry, « Ajustage des Paramètres d'un Régulateur PID »
https://freddy.mudry.org/public/NotesApplications/NAPidAj_06.pdf
https://www.cds.caltech.edu/~murray/courses/cds101/fa04/caltech/am04_ch8-3nov04.pdf
- [3]. Forum arduino « [PID autotuning code, fuzzy logic, Datalogger system heater chauffant 4-20mA](#) »
<https://forum.arduino.cc/t/pid-autotuning-code-fuzzy-logic-datalogger-system-heater-chauffant-4-20ma/>
- [4]. A.Sivert travaux pratique BUT 2 semestre 4 « [systeme-chauffant-pid-4-20ma-iut-geii-soissons](#) »
<https://www.fichier-pdf.fr/2026/02/04/systeme-chauffant-pid-4-20ma-iut-geii-soissons/>
- [5]. C.Grimholt. « Optimal PI-Control and Verification of the SIMC Tuning Rule » 2013
urlr.me/DNdPTtr
https://skoge.folk.ntnu.no/puublications_others/2013_lee-cho-edgar_SIMC-revisited.pdf
- [6]. I. Dogan « PID based pratical digital control » 2022 avec simulateur
<https://www.xlncontrol.com/products/pid/demo.html>
https://www.engineers-excel.com/Apps/PID_Simulator/Description.htm
- [7]. [guide-technique-la-regulation.pdf](#)
<https://sti.eduscol.education.fr/sites/eduscol.education.fr/sti/files/ressources/techniques/677/677-guide-technique-la-regulation.pdf>
- [8]. A.Sivert « A Fuzzy Logic Application for Go-Kart : a Battery Charger » EVER 2011
<https://www.e-kart.fr/images/stories/news/2011/ever/ever11-paper-spbl2.pdf>
A.Sivert « Position control of six-phase induction motor using fuzzy logic: Application to electric power steering » ICEM 2012
<https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6350006>
https://www.espacetechnologue.com/wp-content/uploads/2019/01/TP_Fuzzy.pdf
- [9]. Grafcet programmation langage structuré en C
https://web.enib.fr/~mecatro/automatismes/ST/S2_algo_G7-ST_01-04-2021P.pdf
<https://www.mhj-wiki.de/en/grafcet-studio/device-arduino/>
<https://www.electronique-mixte.fr/grafcet-automatisme/>

<https://circuitdigest.com/microcontroller-projects/arduino-pid-temperature-controller>
<https://github.com/Dhanabhon/AdaptiveFuzzyPID>