**Séance 6 de régulation**

Objectifs de la séance

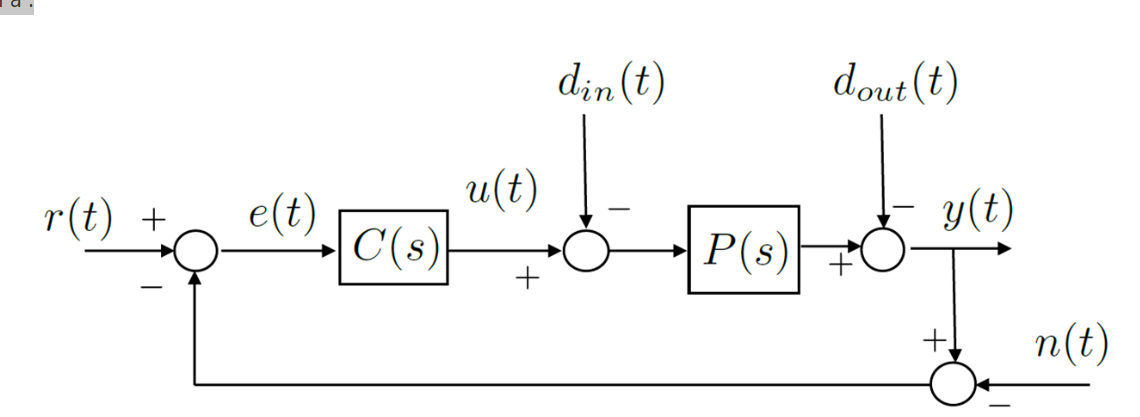
Nous avons à présent vu tout un tas de principes de base utiles pour réguler des systèmes SISO (Single Input Single Output).

Néanmoins, ce qui a été étudié jusqu'à présent n'est applicable qu'au niveau de la théorie, mais certains défis apparaissent si l'on applique ces techniques en réalité. Cette séance abordera donc les défis rencontrés lors de la régulation de systèmes réels et proposera des outils pour les surmonter. Nous conclurons la séance par la présentation de techniques heuristiques permettant de dimensionner rapidement un système réel pour lequel nous ne disposons d'aucun modèle mathématique.

Pour commencer, étudions la prise en compte des perturbations que notre système peut subir et leur impact sur sa réaction !

Perturbations

Si on reprend le schémas général d'un système asservi qui a été présenté lors de la séance d'introduction au cours, on a :

****

Les din(t) et dout(t) représentent les perturbations (disturbances)1 en entrée et en sortie du système et le n(t) (noise)2 représente le bruit sur la mesure. On a appris à dimensionner notre correcteur pour maîtriser la robustesse du système à asservir en observant sa réponse dans le domaine fréquentiel (avec les marges de gain et de fréquence) et/ou en observant les positions de ses pôles (plus leurs parties réelles sont négatives, plus l'amortissement est grand et donc le système robuste). Néanmoins, nous n'avons pas encore prouvé que le système est d'autant plus robuste aux perturbations et/ou bruit que ces marges sont importantes ! Nous avons juste pu voir que la réponse de notre système était d'autant plus amortie que nos marges étaient grandes et on en a déduit que le système était donc plus robuste mais comment vérifier cela en réalité ? 🤔

En fait, nous avons affaire à un système **MISO**(***M***uliple ***I***nput ***S***ingle ***O***utput) contrairement à ce qu'on a eu jusque maintenant où nous n'avions qu'une seule entrée et une seule sortie (→ SISO). Pour résoudre ce problème, nous allons utiliser une technique de résolution que vous connaissez fort bien maintenant après tous ces cours d'électronique : la **théorème de superposition** ! 🤓

On va donc regarder **indépendamment l'impact de chaque entrée** de notre système sur la sortie de ce dernier **et sommer l'ensemble** pour obtenir la fonction de transfert globale de notre système.

Commençons par le plus simple en regardant juste l'impact de r(t) :

1°) On met tous les autres signaux d'entrée à 0 ⇒Une image contenant texte

Description générée automatiquement

2°) On obtient le schéma-bloc suivant :

**Une image contenant diagramme

Description générée automatiquement**

C'est aussi simple que ça ! Je vous laisse donc trouver les fonctions de transfert induites par les 3 autres signaux d'entrée et on se retrouve ensuite à la page suivante

1 Généralement actives dans les basses fréquences comme r(t)

2 Généralement actif dans les hautes fréquences.

**Une image contenant texte

Description générée automatiquement**

***Remarque :*** On a bien un signe "-" qui apparaît aux numérateurs des fonctions de transfert des perturbations et du bruit vu que ces derniers rentrent dans l'entrée négative du différentiateur. De plus, la boucle de retour entrant dans l'entrée positive du différentiateur, il faut également inverser le signal de retour pour pouvoir utiliser la formule de Black. Voici la résolution pour le signal  Din(s):

Une image contenant diagramme

Description générée automatiquement

Maintenant que nous avons toutes ces fonctions de transfert, nous pouvons déterminer la réponse globale de notre système en sommant simplement l'ensemble de celles-ci et en les multipliant par leurs signaux respectifs :

****

Anti-windup

Nous avons vu précédemment qu'il est possible d'utiliser un correcteur avec une action **intégrale** pour rendre l'erreur d'un système finie ou nulle. Cependant, l'ajout d'un tel correcteur n'est pas sans risque ! En effet, le fait que le correcteur incrémente l'erreur au fil du temps peut poser problème si le système n'est pas capable de fournir une sortie suffisante pour compenser cette erreur (= saturation). Dans ce cas, lorsque la consigne est modifiée, le correcteur peut générer une sortie importante et il lui faut du temps pour retrouver une valeur que le système peut effectivement fournir en sortie. Cela peut entraîner des retards plus ou moins longs entre le moment où la consigne est modifiée et le moment où le système réagit.

Un exemple marquant de l'importance de prendre ce problème au sérieux a été vécu il y a quelques années lors du « projet de robot », où le système de régulation des déplacements du robot était activé, mais les étudiants avaient oublié de retirer l'arrêt d'urgence sur les moteurs au préalable. Cela aurait pu entraîner la destruction du robot 😱. En effet, lorsque l'un des étudiants a retiré l'arrêt d'urgence, le robot s'est mis à foncer à toute allure vers le bord de la table et aurait basculé par-dessus si personne n'avait réussi à appuyer sur le bouton d'arrêt d'urgence à temps. Mais pourquoi cela s'est-il produit ?

Eh bien, c'est tout simplement dû au fait que le robot avait reçu l'ordre d'effectuer un premier déplacement au démarrage, et voyant que la position souhaitée n'était pas atteinte (car les moteurs n'étaient pas alimentés), le correcteur intégral a commencé à augmenter la tension à mettre en entrée des moteurs. Ainsi, lorsque l'arrêt d'urgence a été retiré, les moteurs ont immédiatement reçu une tension maximale et ont commencé à tourner à leur vitesse maximale. En soi, cela ne semble pas trop grave, car le correcteur aurait simplement dû diminuer la tension d'entrée des moteurs une fois que l'erreur aurait diminué. Cependant, la commande en tension lorsque le bouton d’arrêt d’urgence a été retiré était supérieure à ce que les moteurs pouvaient recevoir (par exemple, 250% de la tension max). Le correcteur intégral limitait la vitesse à laquelle la commande en tension redescendait, car l’erreur accumulée au fil du temps passé sans bouger était importante. Il fallait donc attendre un certain temps pour que la tension réclamée redescende sous la tension maximale et donc que le robot commence à réellement ralentir. Le problème est que le temps que tout cela se produise, le robot avait depuis longtemps dépassé sa consigne de position et aurait peut-être commencé à reculer seulement 200 à 300 mètres plus loin, ce qui est mortel lorsque la table de jeu ne dépasse pas 3 mètres 😜.

Pour éviter ce problème, il existe pourtant une technique appelée l'anti-windup, qui consiste à limiter la saturation du correcteur intégral en empêchant l'accumulation de l'erreur au-delà de certaines limites prédéfinies. Par exemple, on peut plafonner la commande du correcteur intégral à la valeur maximale admissible par le système, de sorte que la commande ne puisse pas dépasser cette limite, même si l'erreur continue de s'accumuler. Cela permet d'éviter que la sortie du système atteigne sa saturation et de limiter les retards de réponse du système lors de changements brusques de consigne.

Une autre approche consiste à utiliser des techniques de saturation inverse, où l'erreur accumulée est soustraite de la commande du correcteur intégral lorsque la sortie du système atteint sa saturation. Cela permet de réduire l'accumulation de l'erreur au-delà des limites acceptables et d'améliorer la réactivité du système lors de changements de consigne.

Il est important de choisir la technique d'anti-windup appropriée en fonction des spécificités du système à contrôler et des performances requises. Une conception soigneuse du correcteur intégral avec une stratégie d'anti-windup adéquate peut contribuer à améliorer la stabilité, la performance et la sécurité des systèmes de régulation automatique.

En conclusion, bien que le correcteur intégral soit une méthode puissante pour réduire l'erreur en régulation automatique, il peut entraîner des problèmes de saturation et de retard de réponse si les précautions nécessaires ne sont pas prises. L'utilisation d'une technique d'anti-windup appropriée est essentielle pour éviter ces problèmes et garantir le bon fonctionnement du système de régulation.

Afin d'expliquer le problème avec un autre exemple et de présenter la technique de l'anti-windup plus en détails voici une vidéo (in English) traitant du sujet :

Pour parler des problèmes causés par le terme I, on redécompose ce qu’on a appelé « plant » (système) en 2

1. Les actuators qui sont les composants qui génèrent la force ou l’énergie pour changer le système. Ex : un système de chauffage ou moteur.
2. Le process (ce qu’on contrôle avec l’actuator)

Dans la vie réelle, les actuators ne sont pas parfaitement linéaires : ils ne peuvent pas suivre n’importe quelle commande arbitraire qu’on leur donne (il y a des contrecoups, des saturations)

Une image contenant diagramme, schématique

Description générée automatiquement

Imaginons que l’actuator n’est pas capable de suivre la commande donnée.

Par exemple un moteur peut augmenter sa vitesse jusqu’à atteindre sa vitesse max. Mais après, il est impossible d’encore augmenter cette vitesse, même si la commande le souhaite :

Une image contenant diagramme

Description générée automatiquement

Idem pour un baffle ou une pile. En fait aucun composant réel ne suit la consigne tel un système linéaire.

On reprend l’exemple du drône. Le propeller est l’actuator et il réagit à la commande en faisant monter ou descendre le drône via une vitesse. Une image contenant diagramme

Description générée automatiquement

Le propeller génère une force qui soulève le drône dans l’air et change son altitude.

Pour une raison x ou y, on tient le drône au sol. L’erreur vaut 50 m (cste) 🡪 I fait monter violemment la commande pour le faire monter. Si on le retient un moment, l’erreur reste la même (50). La commande, par conséquent, est de faire monter la vitesse pe plus que celles que les hélices peuvent fournir 🡪 on atteint la limite de vitesse des moteurs. La commande augmente toujours, mais on est limité physiquement pour la vitesse des hélices à leur vitesse max.

Une image contenant diagramme

Description générée automatiquement

Le problème survient lorsqu’on veut enlever « l’excès » de commande du terme I.

Imaginons, que le moteur a Nmax= 1000 tr/min et que la commande requiert 2000tr/min. On lâche le drône et il monte à 50m. Puis l’erreur statique diminue. Un fois que le drône est au-dessus de sa commande, l’erreur devient négative et la sortie du terme I (commande) commence à diminuer. Mais il faudra attendre que la vitesse redescende à 1000 rpm pour voir la vitesse diminuer. Pendant ce temps, le drône est parti…

Une image contenant diagramme

Description générée automatiquement

Le but est d’inverser la commande quand l’erreur change de signe. Donc on doit implémenter une sorte de anti-windup méthode.

But des méthodes anti-windup : empêcher la valeur intégrée d'augmenter au-delà d'une certaine limite. Cela répondra directement dans l’autre sens quand l’erreur devient négative.

1. Methode Clamping

Pour éviter la saturation, on peut limiter la sortie du contrôleur PID à la valeur de saturation de l’actuator.

La méthode fait 2 choses : d’abord elle compare la sortie du PID avant et après la vérif en saturation. Si la diff vaut 0 🡪 pas de saturation et la sortie vaut 0. Si ça ne vaut pas 0 🡪 saturation (valeur de 1). Ensuite on compare le signe de la sortie du PID avec le signe de l’erreur. Si les 2 sont +, on sait que le terme I augmente le output pour le rendre plus +. Si les 2 sont négatifs, le système rendra l’erreur encore plus négative.

Donc on regarde si la sortie sature et si l’I a tendance à augmenter le phénomène.

Si le système fait caca, la porte AND passe à 1 et on fixe le terme I à 0. Quand la sortie n’est plus en saturation ou que l’erreur change de signe, alors le terme I est restauré et commence à décroitre.

Une image contenant diagramme

Description générée automatiquement

Sur le drône, dès qu’il atteint sa commande(pas l’altitude ??), l’erreur changera de signe et commencera à réduire la vitesse des hélices, limitant la saturation :

Une image contenant graphique

Description générée automatiquement

NB : en pratique, on limite le système à une vitesse inférieure à sa vitesse max.

### Filtre passe-bas

Comme vous le savez, qui dit mesure dit bruit. Vous avez déjà pu observer son impact lors de vos différents laboratoires d'électronique, en particulier lors de ceux d'électronique analogique où l'utilisation de filtres (passe-bas, passe-haut, passe-bande) est essentielle 🤓.

Les mesures que nous effectuons pour réaliser un retour d'information (feedback) sont souvent de nature électronique et donc sujettes à des perturbations de bruit. Bien que l'amplitude de ces bruits puisse être faible par rapport au signal de consigne, cela peut poser un gros problème lorsque le correcteur que nous utilisons présente une action dérivée. En effet, cette action est d'autant plus importante que la vitesse de variation de l'erreur l'est. Or, les bruits ont tendance à se situer principalement dans la plage des hautes fréquences, créant ainsi des variations rapides dans la mesure et, par conséquent, dans l'erreur ! Pour remédier à cette situation, nous utilisons une solution qui consiste à utiliser un filtre passe-bas pour lisser le signal de mesure, amplifiant ainsi uniquement les variations de l'erreur dues à un changement de consigne et/ou à un choc (théoriquement représenté par un pic de Dirac), et non celles causées par le bruit.

Voici une vidéo (again in English) qui explique cela plus en détail avec des schémas 😉 :

Une image contenant diagramme

Description générée automatiquement

Le bruit peut apparaitre à une f spécifique ou alors sur une bande. Par ex, si le bruit a la même intensité à différentes fréquences, c’est appelé le bruit blanc. Il y a bien d’autres sortes de bruits.

Une image contenant diagramme

Description générée automatiquement

La chose à retenir c’est que les bruits produisent des minuscules décalages de tension sur une large gamme de fréquences ce qui produit à son tour des décalages dans la mesure elle-même.

Des petits bruits mais à haute fréquence peuvent causer des probs.

Ce que le capteur peut mesurer peut être une belle quantité lisse comme une température qui augmente lentement mais en raison du bruit de capteur, la mesure peut être irrégulière.

Une image contenant diagramme

Description générée automatiquement

Ce n’est pas toujours un problème mais dès qu’il y a un terme dérivé dans le PID ça en est un car il est capable d’amplifier de faibles bruits.

Exemple avec un sin en faible et haute fréquence :

Slope = pente et steeper est « +raide »

Une image contenant graphique

Description générée automatiquement

On peut réduire l’A du bruit et revenir à une pente similaire à celle en BF :

Une image contenant graphique

Description générée automatiquement

Si on aumente f :

Une image contenant texte

Description générée automatiquement

Cela se démontre mathématiquement : Une image contenant texte

Description générée automatiquement

On veut un filtre qui empêche les f au-dessus d’un certain point d’entrer ds notre dérivée afin d’éviter des problèmes. Ce point est la fréquence de coupure.

On a le choix de la placer où on veut (par exemple le filtre vert ou orange) :

Une image contenant diagramme

Description générée automatiquement

On ajoute le composant « filtre au système » :

Une image contenant diagramme

Description générée automatiquement

En réalité, tout le bruit ne peut pas être enlevé et tout le signal utile ne peut pas passer à travers le filtre. Heureusement, bcp d’applications ont des bruits de puissance et d’amplitude faibles.

Svt, les systèmes étudiées sont de BF et de forte puissance 🡪 on utilise un filtre passe-bas pour bloquer ou atténuer les amplitudes de HF.

Une image contenant graphique

Description générée automatiquement

Evidemment, tout le bruit ne peut pas être enlevé et donc le but c’est de trouver l’endroit où placer la fréquence de coupure afin d’atténuer fortement le bruit tout en gardant les fréquences du signal utile.

Une image contenant texte

Description générée automatiquement

Cutoff = fréquence de coupure.

Petite transformation de formule :

Une image contenant texte

Description générée automatiquement

Exrpession du filtre + dérivée dans le domaine de Laplace :

Une image contenant diagramme, schématique

Description générée automatiquement

Une autre manière de le voir :

Une image contenant diagramme

Description générée automatiquement

C’est intéressant car on a 2 choix : un filtre passe-bas en série avec une dérivée ou alors une boucle de rétroaction avec une intégrale dans la boucle de rétroaction.

La première méthode est plus facile à lire, à comprendre.

La deuxième est plus efficace pour les algorithmes.

### Contrainte de rapidité des systèmes en cascade

La **régulation en cascade** d'un système consiste à contrôler la sortie d'un système composé de plusieurs sous-systèmes en série, en agissant sur l'entrée du premier sous-système. Bien que ce type de régulation présente des avantages tels que la simplicité de mise en œuvre et la réduction des perturbations, il comporte également des **inconvénients**tels que la **dégradation des performances** et l'**instabilité**.

Pour rappel, lors de la deuxième séance théorique, nous avons déjà étudié un exemple de système en cascade avec un système de gestion de la position d'un moteur électrique (cf. démonstration faite lors de la première séance de cours) :

Une image contenant diagramme

Description générée automatiquement

Légende :Schéma fonctionnel d’un système de contrôle de position. Le système dispose de trois boucles en cascade pour le contrôle du courant, de la vitesse et de la position. Chaque boucle a une valeur de référence fournie en externe (indiquée par l’indice 'r') qui définit la valeur nominale de l’entrée à la boucle, qui est ajoutée à la sortie de la boucle externe suivante pour déterminer la valeur commandée pour la boucle (appelée « point de consigne »). [Source : Aström, Feedback Systems - chap. 1.5, p. 1-17]

Les problèmes liés à la régulation en cascade sont principalement dus aux interactions entre les sous-systèmes et aux retards de transmission. En effet, chaque sous-système introduit un déphasage et une atténuation du signal de sortie par rapport au signal d'entrée, ce qui affecte la précision et la rapidité de la régulation. De plus, les retards de transmission entre les sous-systèmes entraînent des décalages temporels entre les actions correctives et les effets observés, ce qui peut provoquer des oscillations ou des divergences du système.

Pour éviter ces problèmes, il est important de choisir soigneusement les paramètres de réglage des régulateurs associés à chaque sous-système, en tenant compte des caractéristiques dynamiques et statiques de l'ensemble du système. Il est donc essentiel de s'assurer que les critères de performance des sous-systèmes sont suffisamment bons pour que le système global ne soit pas perturbé par leur fonctionnement. Reprenant l'exemple ci-dessus de la position d'un moteur électrique, il est nécessaire que la boucle d'asservissement du courant stabilise rapidement le courant par rapport à la boucle de régulation de la vitesse, de sorte que cette dernière ait l'impression que la consigne de courant génère directement le courant souhaité. Il en est de même pour la boucle d'asservissement de la vitesse par rapport à celle de la position.

Cependant, obtenir les performances requises pour chaque sous-système afin d'assurer le bon fonctionnement de l'ensemble du système peut être complexe. Il peut être nécessaire de recourir à des techniques de régulation avancées, telles que la régulation prédictive ou la régulation adaptative, pour améliorer les performances du système. Ces techniques ne seront néanmoins pas abordées dans ce cours.

Systèmes digitaux

De nos jours, de nombreux moyens de régulation des systèmes utilisent des outils numériques, tels que des microcontrôleurs, pour contrôler la performance des systèmes. Contrairement aux approches analogiques continues, ces systèmes numériques utilisent des valeurs échantillonnées à des intervalles de temps réguliers, ce qui peut poser des défis spécifiques.

Un problème courant dans la régulation numérique est que la fréquence d'échantillonnage peut avoir un impact significatif sur la stabilité du système. Si la fréquence d'échantillonnage n'est pas suffisamment élevée par rapport à la dynamique du système, cela peut entraîner des instabilités et des performances sous-optimales. En effet, les correcteurs PID et autres algorithmes de régulation sont généralement conçus en se basant sur des modèles continus, et leur comportement peut être différent lorsque des échantillons discrets sont utilisés.

Pour éviter ces problèmes, il est essentiel de prendre en compte la fréquence d'échantillonnage dans la conception des correcteurs PID pour les systèmes numériques. Cela peut nécessiter la discrétisation du modèle du système, c'est-à-dire la conversion du modèle continu en un modèle discret qui tient compte de la fréquence d'échantillonnage. Ensuite, les gains du correcteur PID peuvent être ajustés en conséquence pour garantir la stabilité et la performance du système dans le domaine discret.

Il est important de noter que la fréquence d'échantillonnage doit être choisie judicieusement en fonction des caractéristiques du système à contrôler, et que des méthodes d'analyse et de conception spécifiques aux systèmes numériques doivent être appliquées pour garantir une régulation efficace et stable. Les ingénieurs en contrôle de systèmes doivent donc être conscients de ces défis lorsqu'ils conçoivent et mettent en œuvre des systèmes de régulation numérique, tels que ceux basés sur des microcontrôleurs, des DSP (*Digital Signal Processor*) ou des automates programmables industriels.

Pour mettre en avant les problèmes liés à la mise en cascade et à la discrétisation du modèle d'un système, je vous invite à visionner la vidéo (*still in English*) suivante :

Qu’est-ce qu’un contrôle en cascade ?

On reprend le DRONEEEE

Une image contenant diagramme

Description générée automatiquement

On se focus dur le propeller. La commande est l’entrée et la vitesse résultante est la sortie.

Askip c’est remplacé par :

Une image contenant diagramme, schématique

Description générée automatiquement

Le outer loop conduit le set point de la loop intérieure et la loop intérieur inluence la boucle de contre réaction de la boucle ext 🡪 les 2 sont connectées.

Pourquoi utiliser des boucles en cascade ?

Pourquoi ne pas utiliser cette boucle par ex : le contrôleur prend l’erreur en altitude et en ressort une tension 🡪 l’altitude influence direct la tension.

1. Car c’est plus simple d’isoler les problèmes avec la méthode de boucles en cascade. Ex : si prob avec le moteur : ce sera difficile de tester juste le moteur séparément du reste. Avec la bocule sur le moteur simplement : on peut vérif si le problème vient direct du moteur ou si c’est le reste du système avec lequel il interagit où il y a un prob.
2. Différents groupes peuvent travailler sur les différentes boucles.
3. Plus important : faire fonctionner des boucles en cascade permet de les faire fonctionner à différentes vitesses pour résoudre différents problèmes et sources d’erreur. Le contôleur du moteur peut réagir rapidement aux perturbations locales alors que le contrôleur en altitude peut être réglé pour rejeter le bruit du capteur et augmenter la stabilité .

Exemple : drône a une U de 10V et une rpm de 100tr/min. Imaginons que la batterie lache( pas vrmnt ça anywayy), la boucle du moteur est sensible à cette perturbation et ajuste le voltage en une fraction de seconde. Donc les hélices sont barely affectées. Si la boucle intérieure est assez rapide, alors la boucle externe ne verra même pas les perturbations parce que il n’y aura pas de changement notables concernant l’altitude. Cela permet à la boucle externe d’être plus lente et de répondre seulement à des perturbations lentes (wind gusts ?)

Avec une seule boucle, le controleur de l’altitude devrait avoir une sensibilité aux petites perturbations de l’altitude et changer rapidement la tension du moteur. Si on la tune pour qu’elle soit plus agressive, cela répondra plus rapidement aux changements en altitude mais cela répondra également plus rapidement aux bruits… Et c’est pas idéal.

Comment régler la cascade de boucles ?

Si la boucle int est plus rapide que la boucle ext ( la bande passante sera plus grande d’au moins 7x), on peut les tuner séparément.

On peut commencer par designer la boucle intérieure, la rendre rapide à tel point qu’on pourrait dire que la boucle intérieure n’existe pas (adaptation instantanée du système à la bonne vitesse). Pour tuner la boucle ext, on pourra ainsi le faire en ne tenant pas compte de la boucle int :

Une image contenant diagramme, texte

Description générée automatiquement

Si les 2 doivent avoir la même bande passante et donc avoir la même rapidité, on ne peut pas affirmer que la boucle intérieure est assez rapide que pour être instantanée. 🡪 la boucle interne influence la boucle externe.

Il y a plusieurs méthodes pour modéliser :

1. La méthode itérative consiste à modéliser la bocule int avec une supposition puis régler la boucle externe pendant que la boucle interne est en cours d’exécution. Si cela ne suffit pas pour les 2 boucles, lorsque vous avez terminé, vous revenez en arrière et modifiez la boucle int et puis on réittère jusqu’à a voir les perfos voulues.
2. Une manière simple est de traiter le système comme un système multi-entrées et multi-sorties plutôt que deux systèmes en cascade à une seule entrée et une seule sortie. Ca permet de traiter l’ensemble comme un système dynamique unique et de régler les 2 boucles simultanément en utilisant une variété de réglages basés sur les méthodes de réglage.

Une image contenant texte, plein air

Description générée automatiquement

1. On peut utiliser des logiciels

**PID DISCRET**

Les systèmes étudiés sont généralement exécutés sur ordinateur et ces derniers ne fonctionnent pas en continu : ils prennent des échantillons à une fréquence régulière (1x/h par ex). Ex si le logiciel fonctionne à 1Hz alors une fois par seconde, le contrôleur lit les capteurs, effectue les calculs et commandes les actionneurs qui gardent la même consigne jusqu’à la prochaine commande (🡪 1s après).

Parlons du temps d’échantillonnage. Si ce dernier est faible par rapport à la dynamique du système, il se comporte comme un système continu. Le contrôleur lit et met assez rapidement les actionneurs à jour que ça apparait continu. Si le temps d’echantillonnage est élevé, alors le PID voit tout et réagit juste comme avant. S’il est faible, le contrôleur sera également ralenti pour l’empêcher de faire du caca.

La structure algorithmique du PID est également différente. KI,KP et KD ne sont pas idem mêmes pour des systèmes identiques. Il y a différents méthodes pour convertir du domaine s en domaine z.

Une image contenant diagramme

Description générée automatiquement

Comment passer du s au z ? (continu à discret) :

Une image contenant texte

Description générée automatiquement

Pourquoi passer par des systèmes continus avant de passer par le système discrétisé ?

C’est la même question que de passer par des systèmes linéaires avant d’aller vers des systèmes non linéaires.

Parce que résoudre des problèmes dans s est plus facile et on a bcp d’outils à notre disposition pour les domaines continus. En plus, le système devant être contrôlé est lui-même continu donc modéliser le système et le contrôleur dans le même domaine continu rend le problème + simple.

Méthode heuristiques

En pratique, il n'est pas toujours possible de disposer d'un modèle mathématique fiable du système à réguler, mais seulement de mesures sur celui-ci. Nous avons vu qu'il est possible d'identifier un modèle mathématique à partir de ces mesures, mais cela peut prendre du temps et les critères de performance peuvent ne pas justifier cette démarche, notamment pour des systèmes déjà naturellement stables avec peu de dépassement. Par conséquent, lorsque l'on souhaite réguler rapidement un système, on a recours à des méthodes pragmatiques qui ont fait leurs preuves pour certaines catégories de comportements dynamiques des systèmes asservis (ex: critère de Ziegler-Nichols ou celui de Aström-Hägglund). Ces **méthodes heuristiques** permettent de trouver rapidement des paramètres plus ou moins corrects d'un correcteur PID pour obtenir un fonctionnement acceptable. Pour en savoir plus, je vous invite à consulter les points 8.1 et 8.2 du syllabus de Mme Vetcour.

8.1 Identification

Dans certains cas, il n'est pas possible de déterminer facilement un modèle analytique de l'installation à régler. On n'a donc pas une connaissance mathématique, mais il faut bien avoir une certaine connaissance du système pour dimensionner son régulateur. Celle-ci est obtenue par une mesure sur l'installation réelle. La plus simple à mettre en œuvre est l'essai indiciel.

8.1.1 Essai indiciel

On observe la réponse indicielle du système (réponse à un changement brusque de consigne). Pour une réponse en forme de S :

Une image contenant diagramme

Description générée automatiquement

On détermine le point d'inflexion (+) sur la réponse indicielle, et la tangente en ce point: les intersections de celle-ci avec l'axe du temps et avec la valeur asymptotique de la réponse indicielle permettent de définir deux temps caractéristiques τm (temps mort) et T. Sur une mesure réelle, le tracé "manuel" de la tangente au point d'inflexion est très imprécise, et peut conduire à de très grandes variations de τm .

τm prend en compte, au moins approximativement, des interactions des différents processus d’accumulation ainsi que les différents phénomènes éventuels de transports de matière au sein de l’installation. L'instant du point d'inflexion correspond à celui du maximum de la dérivée de la réponse indicielle (en pointillé sur la figure) et la valeur du maximum correspond par définition à la pente en ce point, donc 1/T . On aura donc avantage à mesurer la réponse avec un système d'acquisition de données – qui permet de calculer numériquement la dérivée – plutôt qu'avec un simple oscillographe. Les temps caractéristiques τm et T s'obtiennent alors par simple calcul géométrique. Les critères décrits aux paragraphes suivants s'appliquent bien pour des systèmes mesurés sans dépassement ou à dépassement très faible, faute de quoi les résultats obtenus seront assez loin des attentes. Remarquons que certains systèmes ne peuvent pas supporter un tel essai ; on leur appliquera donc un autre essai typique, en boucle fermée avec un simple amplificateur de gain g ajustable.

8.2 Critère de Ziegler-Nichols (1942)

Une image contenant texte

Description générée automatiquement

Une image contenant texte

Description générée automatiquement