**Régu séance 5**

Objectifs de la séance

Dans ce cours, nous avons appris à analyser les caractéristiques d'un système en boucle fermée, à l'aide d'outils tels que la fonction de transfert (paramètres des formes canoniques d'un système du 1er et du 2nd ordre), la réponse indicielle, les diagrammes fréquentiels (Bode, Nyquist et Nichols) ou le lieu des racines (pôles et zéros). Ces outils nous permettent de déterminer si un système est stable ou non, et quelles sont ses performances en termes de précision, de rapidité ou de robustesse.

Mais comment faire si le système ne répond pas à nos exigences ? Comment modifier son comportement pour l'améliorer ? C'est là qu'intervient le correcteur. Un correcteur est un dispositif qui agit sur la commande du système asservi pour réduire l'écart entre la sortie mesurée et la consigne désirée. Il existe différents types de correcteurs, mais nous allons nous intéresser au plus courant : le **régulateur PID**.

Le régulateur PID

Le **régulateur PID** est composé de trois actions : proportionnelle (P), intégrale (I) et dérivée (D). Chacune de ces actions a un effet différent sur le comportement du système asservi :

* L'action **proportionnelle** (**P**) produit une **correction proportionnelle à l'écart entre la valeur mesurée et la valeur désirée**. Elle permet de réduire rapidement cet écart, mais elle peut aussi provoquer des oscillations ou un dépassement de la consigne.
* L'action **intégrale** (**I**) produit une **correction proportionnelle à l'intégrale de l'écart dans le temps**. Elle permet d'augmenter la classe du système (=> éliminer par exemple l'erreur statique si le système est de classe 0). Elle peut aussi augmenter le temps de réponse ou la sensibilité aux perturbations.
* L'action **dérivée**(**D**) produit une **correction proportionnelle à la dérivée de l'écart par rapport au temps**. Elle permet d'anticiper les variations futures de l'écart et d'amortir les oscillations ou les dépassements. Elle peut aussi amplifier le bruit ou les variations rapides du signal mesuré.

Un régulateur PID est donc un compromis entre ces trois actions, qui doivent être réglées selon les caractéristiques du système asservi et les performances souhaitées. Il existe des méthodes empiriques ou analytiques pour déterminer les paramètres **optimaux** du régulateur PID.

Un régulateur PID est très utilisé dans l'industrie et dans divers domaines d'application, car il présente plusieurs avantages :

* Il est **simple** à mettre en œuvre et à comprendre.
* Il est **robuste** face aux variations des paramètres du système asservi ou aux perturbations extérieures.
* Il peut s'adapter **à différents types de systèmes linéaires** ou **non linéaires**, continus ou discrets, monovariables ou multivariables.
* Il peut offrir des performances satisfaisantes pour des objectifs variés : stabilité, précision, rapidité, etc.

Afin de bien comprendre l'utilité et la manière donc chaque paramètre d'un correcteur PID influence le comportement d'un système asservi, voici une vidéo (*in English*) présentant ce correcteur d'une manière intuitive :

Une image contenant diagramme

Description générée automatiquement

Plant est le système que l’on veut contrôler ou bien le système dont on veut affecter le comportement.

Actuating signal = signal d’actionnement (aussi appelé plant input)

L’idée de base d’un « control system » est de déterminer comment générer un le bon « plant input » pour que le système produise le bon plant output.

Job de l’ingé : produire le bon input au système pour obtenir la sortie que l’on souhaite.

Commanded varaible = setpoint = reference = desired value

Feedback permettant de savoir à quel point la sortie du système est loin de ce qu’on veut qu’elle soit.

La différence entre la commande et le feedback est l’erreur.

La question est comment est ce que à partir de l’erreur, on la converti en une commande adaptée à « l’actionneur » et ainsi amener l’erreur à 0. 🡪 Avec un contrôleur.

Exemple : imaginons que l’on est dans le goal d’un terrain de foot et qu’on veut aller au milieu du terrain qui est 50 m plus loin. Dans ce cas nous somme le « plant (système que l’on veut controler)».

Le actuating signal ou signal d’actionnement sont notre vitesse et la direction vers laquelle on marche. La sortie est la distance à laquelle on est.

Au départ, la sortie est à 0m et l’entrée à 50m

L’erreur de position au départ est de 50 m. (50-0)

Le contrôleur c’est notre cerveau qui commande la vitesse des jambes.

Et une façon que le cerveau a de faire est d’utiliser l’erreur au moment présent pour décider de la vitesse de marche. Par exemple ici, si on met le controller à 0.1, ça signifie que on prend l’erreur du système et on la multiplie par 0.1, on aura notre vitesse de marche. La vitesse au début sera de 5m/s vu que l’erreur est de 50-0=50 m.

Une image contenant diagramme

Description générée automatiquement

On parle de contrôleur proportionnel. Quand nous sommes loin du milieu du terrain (au début quoi), on réduit l’erreur rapidement et au fur et à mesure que l’on s’approche de l’objectif (ici le centre), l’erreur diminue moins rapidement.

Et donc on avance rapidement au début puis de moins en moins vite.

De cette façon , on atteindrait finalement et asymptotiquement les 50m, u moment où l’erreur serait nulle et le contrôleur proportionnel multiplierait cela par 0.1 ce qui donnerait une vitesse de 0 m/s.

Une image contenant diagramme

Description générée automatiquement

Une image contenant texte, ligne, Police, capture d’écran

Description générée automatiquement

Si nous voulions ajuster le temps qu’il nous a fallu pour atteindre l’objectif, on pourrait augmenter ou diminuer le terme multiplicatif.

Avec ce type de contrôleur, quelle que soit le terme multiplicatif choisi, on sait qu’on s’arrêtera sur la ligne de but tout juste.

Ca semble génial… 2eme exemple

Imaginons qu’on veut designer un contrôleur d’altitude d’un drone quadricoptère : (4hélices) (similaire à l’exemple du foot puisque on essaye d’aller à une certaine position)

Le système que l’on veut contrôler est le drone (plant) et l’output est l’altitude

Une image contenant diagramme

Description générée automatiquement

Avec ce contrôleur, au début c’est nickel, la vitesse des hélices augmente fort ce qui est normal vu que l’erreur est grande

Une image contenant texte

Description générée automatiquement

A 50m, l’erreur est nulle 🡪 les hélices s’arrêtent 🡪 le drône tombe…

Quand cela arrive, la vitesse de l’hélice recommence à augmenter car l’erreur augmente.

Il y a une certaine vitesse pour laquelle la force de poussée est égale au poids du drône 🡪 le drône va donc rester à cette vitesse d’hélice et cette position.

(à quelle distance il va donc rester de l’objectif ?) Ce qui va fixer la distance à laquelle le drone va planer est donc cette vitesse qui est commandée par le facteur multiplicatif de l’erreur.

Si le gain du contrôleur vaut 2 et que la vitesse de rotation où la force de poussée compense le poids du drone est de 100, alors le drône se stabilise au sol. Puisque l’erreur vaut 50 et multiplié par 2 ça fait 100 rpm et du coup pas assez pr monter.

Une image contenant texte

Description générée automatiquement

Avec un gain de contrôleur de 5 Une image contenant diagramme

Description générée automatiquement

Peu importe, à quel point nous avons augmenté le gain, l’erreur ne disparaitra pas. Elle deviendra de + en + petite avec ce contrôleur mais elle ne disparaitra jamais.

Une image contenant diagramme

Description générée automatiquement

Conclusion : contrôleur proportionnel est bien mais pas pour tous les systèmes car il crée une erreur constante pour le drone = erreur d’état stable.

Comment enlever cette erreur en régime permanent ? Nous pouvons laisser le contrôleur utiliser les informations passées en ajoutant spécifiquement un chemin d’intégrateur dans notre contrôleur qui est // au contrôleur prop.

Une image contenant diagramme

Description générée automatiquement

Un intégrateur traduit le signal d’entrée dans le temps en conservant le total cumul 🡪 il a en mémoire ce qu’il s’est passé avant. Si le drône occupe mtn une position stable en dessous de la hauteur demandée (ici 50 m), le terme de l’erreur est non nul et quand il est intégré la sortie augmentera tant qu’il y aura une erreur ds le système. Et cette valeur qui augmente provenant du terme intégrateur va permettre d’augmenter la vitesse des hélices et donc permettre au drone de continuer à monter.

Une image contenant diagramme

Description générée automatiquement

Une image contenant diagramme

Description générée automatiquement

A 50 m :

Une image contenant diagramme

Description générée automatiquement

Le terme proportionnel ne rajoute aucune vitesse puisqu’il fait erreur \* gain or erreur vaut 0.

Mais le « chemin » intégrateur a additionné et soustrait les valeurs des erreurs jusqu’à arriver à 100 rpm. En effet, par ex qd on est au steady state error et que le terme proportionnel qui ne rajoute plus de vitesse, on a tjs une erreur qui va venir augmenter la partie integral et ce jusqu’à ce qu’on dépasse les 50 metres et ainsi avoir qqch de negatif et au final des oscillations pe de plus en plus petites qui vont tendre vers les 100 rpm au niv des 50 metres, Après, la sortie de l’intégrateur ne bougera plus puisque on a 0 en entrée.

Ça correspond au controleur PI : qqch qui comprends le présent et qui a une mémoire du passé

On a donc atteint le but mais le chemin du drone pr y arriver pourrait ne pas être idéal.

Imaginons qu’on soit juste en dessous des 50 m (une situation intéressante peut apparaitre) :

Une image contenant diagramme

Description générée automatiquement

Ds ce cas, la branche proportio est 0 puisque à ce point, l’erreur est très petite. Et dépendant de comment le drone a été à cette hauteur, l’intégrateur a pu sommer jusqu’à des valeurs au delà de 100 rpm ce qui signifie que le drone va continuer à monter

Si l’intégrateur dépasse les 100 rpm pour arriver aux 50 mètres, il va monter ce qu’on veut puisqu’il n’a pas encore atteint les 50m. Cependant, pour compenser cet excès de vitesse, il devra aller un peu plus haut que 50m pour créer une erreur négative. En additionnant avec cette erreur négative l’erreur de base, la vitesse diminue.

On ajoute une branche pour corriger cet effet qui peut ne pas être voulu.

Une image contenant diagramme

Description générée automatiquement

On ajoute une branche qui va pouvoir prédire le futur et dire à quel point on va bientôt atteindre notre objectif.

Une dérivée mesure le taux de variation de l’erreur c’est-à-dire la vitesse à laquelle l’erreur augmente ou diminue.

Par exemple, si le drône monte rapidement et s’approche rapidement de l’objectif, cela signifie que l’erreur diminue rapidement. Cette erreur décroissante a un taux de changement négatif, ce qui produira une valeur négative à travers notre chemin de dérivées. Cette valeur négative sera ajoutée à la sortie des contrôleurs donc la vitesse de l’hélice baisse.

Le contrôleur utilise une valeur de changement pour déterminer que nous nous rapprochons trop rapidement de notre objectif puis ralentissons prématurément la vitesse de l’hélice, empêchant le drône de dépasser.

C’est un controller polyvalent qui utilise l’erreur actuelle, l’erreur passée et une prédiction de la future erreur pr calculer la commande appropriée à l’actionneur.

Une image contenant diagramme

Description générée automatiquement

Impacts des différents paramètres d'un régulateur P

Concernant le correcteur proportionnel **P**, vous avez normalement pu observer :

**En régime statique,** une augmentation du gain entraîne une **diminution de l’erreur**.

**En régime dynamique,** cette augmentation de gain rend le système **plus rapide** mais, au-delà d’une valeur limite, elle **augmente l’instabilité** du système.

L’effet sur la **réponse fréquentielle** se traduit par une **translation en module de la réponse**.

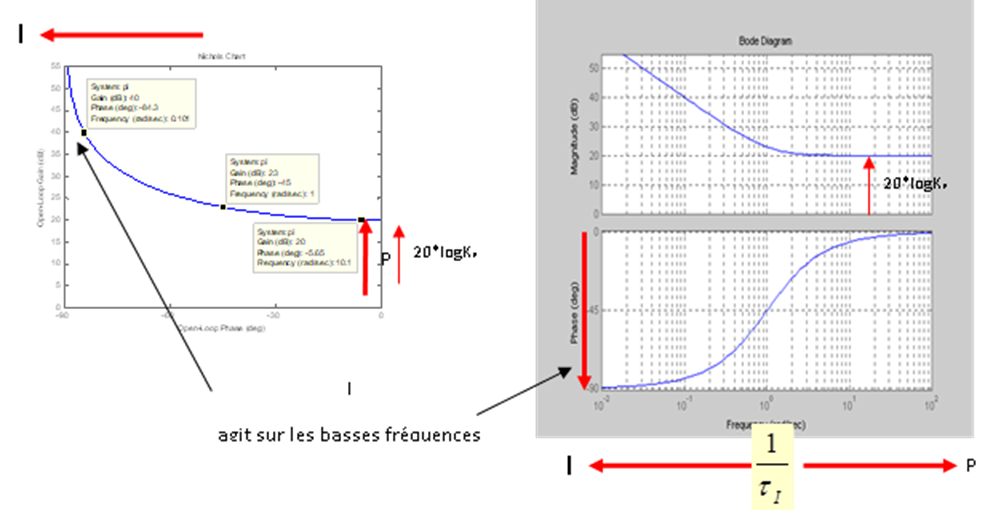
**L’inconvénient**du correcteur proportionnel est son **incapacité à annuler l’erreur en régime permanent** (d’un système de classe 0 ne possédant aucun pôle à l’origine) lorsque la grandeur d’entrée est de type échelon (erreur statique).

Impacts des différents paramètres d'un régulateur PI

Concernant le correcteur proportionnel **PI** :  
  
**Pour**t→∞ (**régime statique** ou permanent) : l’effet du terme intégral est prépondérant : il **annule l’erreur statique**.

**Pour**t→0 (**régime dynamique** ou transitoire) : nous avons vu que le terme intégral avait pour inconvénient d’**augmenter le temps de réponse** (système moins rapide), et d’**augmenter l’instabilité** (introduction d’un déphasage supplémentaire de -90°). Le rôle de la partie P du correcteur PI est de remédier à ces inconvénients.

Par ailleurs, le terme intégral permet de **filtrer la variable à régler** d'où l'utilité pour le réglage des variables bruitées telles que la pression.



Impacts des différents paramètres d'un régulateur PD

Les effets que présente un correcteur **PD** sont les suivants :

* **En régime dynamique,** l’intérêt principal de la correction dérivée est de s’opposer aux grandes variations de l’erreur (donc aux oscillations) puisque le correcteur D modifie la grandeur de réglage en fonction de la vitesse de variation du signal d’erreur (en pratique, l’impulsion est limitée à la constante de temps τD). Elle permet donc de **stabiliser** le système et d’**améliorer le temps de réponse**.
* **En régime statique** (entrée en échelon ou évolution constante) : le D n’intervenant que sur la dérivée de l’erreur, si l’erreur est constante, le dérivateur n’a aucun effet, c’est donc le terme P qui agit en régime établi.

Une image contenant texte, capture d’écran, diagramme, ligne

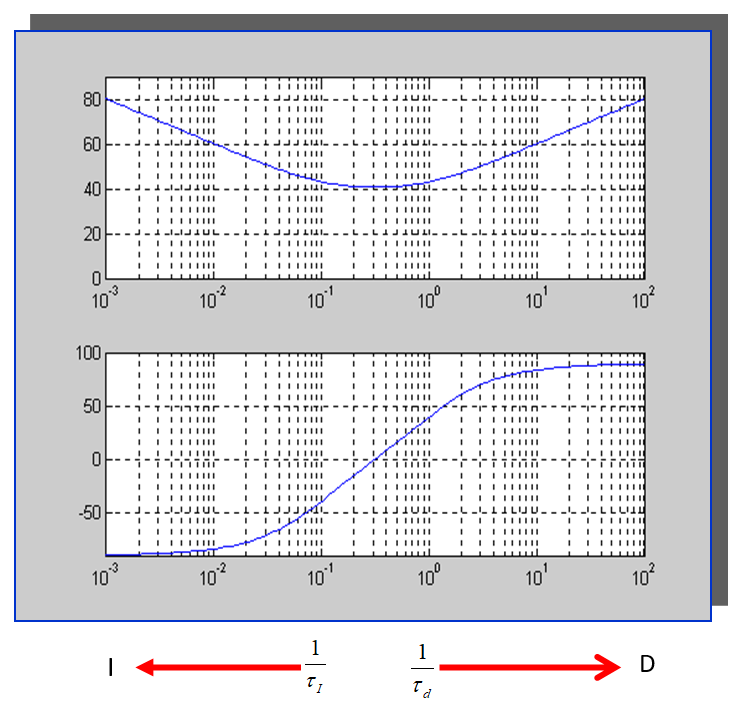
Description générée automatiquement

**Problèmes liés à l'action dérivée**

* L’action dérivée permet de compenser les inerties dues au temps mort, d’**accélérer la réponse du système**et d’**améliorer la stabilité de la boucle**. Elle contribue notamment à **amortir rapidement les oscillations** provoquées par une perturbation ou une variation soudaine de la consigne. L’action dérivée est utilisée dans l’industrie pour le réglage des **variables lentes**, comme la température. Elle n’est **pas recommandée** pour le réglage d’une variable bruitée ou trop dynamique, comme la pression. En effet, en dérivant un bruit, on **risque d’augmenter son amplitude au point de masquer le signal utile**.
* Souvent, la consigne varie par paliers (type échelon) et donc le signal d’erreur change également par sauts brusques. L’effet du terme dérivé est alors théoriquement infini, ce qui peut saturer le régulateur et le rendre insensible aux variations du signal. En pratique, on préfère souvent lier l’action du dérivateur **aux variations de la grandeur à régler seule** et non à l’écart mesure-consigne (erreur), afin d’éviter les à-coups dus à une variation subite de la consigne.

Le correcteur **PID** se comporte :

* Pour les **basses fréquences** comme un **intégrateur**donc le système sera précis d’un point de vue statique.
* Pour les **hautes fréquences** comme un **dérivateur** qui introduit une avance de phase de +90° donc une amélioration de la stabilité.
* Il a donc pour effet (sur toutes les fréquences) d'**annuler l'erreur statique** (en augmentant la classe du système), de **stabiliser** le système et d'**augmenter sa rapidité**.



Fin de la séance

Bravo ! Vous avez terminé la séance théorique ! 🥳

Vous avez appris :

* Ce que sont les correcteurs P, PI, PD et PID ;
* Quels sont les effets de chaque paramètre sur la réponse d’un système ;
* Les avantages et les inconvénients de chacun des correcteurs.

Vous disposez maintenant de tous les outils nécessaires pour commencer à réguler des systèmes ! 😎

Il est temps de passer à la pratique et de réguler des systèmes selon des critères définis en choisissant le correcteur approprié et en le dimensionnant lors du prochain labo et des suivants ! 🤓