

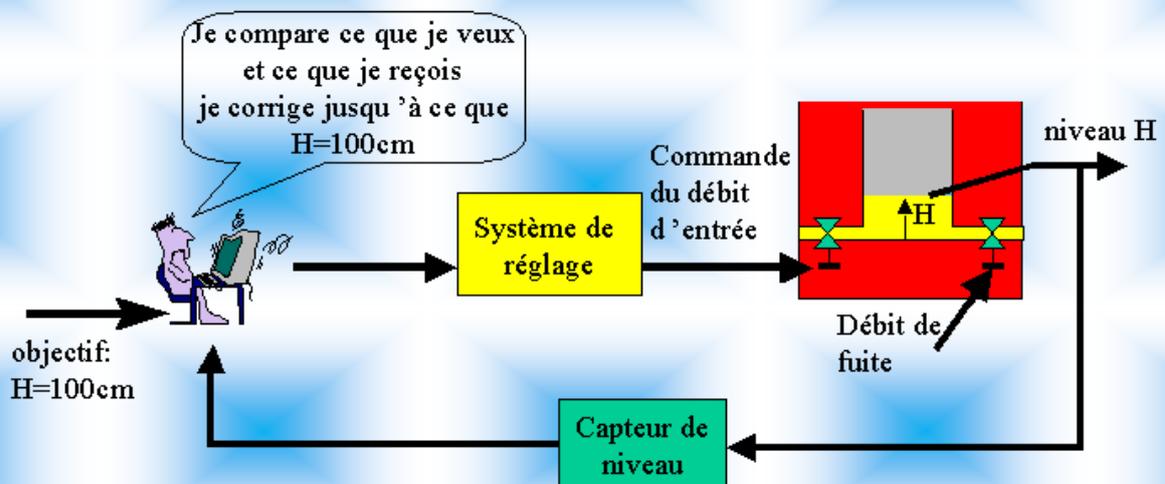


*Support de Cours, Exercices Corrigés et Scripts en Matlab
Destiné aux étudiants en Licence Electromécanique*

REGULATION INDUSTRIELLE

Réalisé par : Dr. Moussa ATTIA^{1,2}

Maître de Conférences à l'Université de Tébessa



¹moussa.attia@univ-tebessa.dz

²<https://orcid.org/0000-0003-3700-7049>

Table des matières

Avant-propos	1
Les Chapitres	
1 Introduction à la régulation industrielle	2
1.1 Introduction	2
1.2 Définition	3
1.3 Analyse des opérations d'une chaîne régulation	3
1.4 Qu'est-ce qui constitue une boucle de contrôle?	4
1.5 Boucle ouverte et boucle fermée	5
1.5.1 Boucle ouverte	5
1.5.2 Boucle fermée	6
1.6 Exemples de système de régulation (Véhicule électrique)	7
1.7 Conclusion	8
2 Régulateur tout ou rien	9
2.1 Introduction	9
2.2 Systèmes tout ou rien	9
2.3 Systèmes à hystérésis	12
2.4 Caractéristiques complexes	14
2.5 Conclusion	15
3 Identification des systèmes en boucle ouverte et fermée	16
3.1 Introduction	16
3.2 But de l'identification	16
3.3 Choix du modèle	17

3.4	Identification en chaîne ouverte	17
3.4.1	Courbe en S	21
3.4.1.1	Méthode de Strejc	21
3.4.1.2	Méthode de Strejc-Davoust	23
3.4.1.3	Méthode de Broida	24
3.4.2	Courbe intégratrice	25
3.4.3	Courbe avec oscillations	27
3.5	Identification en chaîne fermée	28
3.5.1	La méthode des oscillations limites	28
3.5.2	Premier essai	29
3.5.3	Deuxième essai	30
3.6	Conclusion	31
4	Les régulateurs standards : P, PI, PD et PID	32
4.1	Introduction	32
4.2	Principe général de la correction d'un système	32
4.3	Actions correctives élémentaires	33
4.3.1	Correcteur proportionnel	33
4.3.2	Correcteur intégral	34
4.3.2.1	Définition	34
4.3.2.2	Influence sur les autres performances	34
4.3.3	Correcteur à action dérivée	36
4.4	Inconvénient fondamental des actions correctives élémentaires	39
4.5	Action proportionnelle intégrale correcteur à retard de phase	40
4.5.1	Définition	40
4.5.2	Exemple	41
4.6	Action proportionnelle dérivée correcteur à avance de phase	43
4.6.1	Définition	43
4.6.2	Exemple	45
4.6.3	Stratégie de réglage d'un asservissement	47
4.7	Correcteur Proportionnel Intégral Dérivé, PID	47
4.7.1	Structure des correcteur PID	48
4.7.1.1	Structure série	48
4.7.1.2	Structure parallèle	48
4.7.1.3	Structure mixte	49
4.7.2	Méthodes de synthèse	49
4.7.2.1	Méthodes empiriques	49

4.7.2.2	Méthodes algébriques	52
4.8	Conclusion	57
5	Choix et dimensionnement des régulateurs	58
5.1	Introduction	58
5.2	Réglage par critère temporel	58
5.2.1	Réglage PID par modèle de référence	59
5.2.2	Réglage de Naslin	63
5.2.3	Réglage P, PI et PID par les méthodes de Ziegler-Nichols	64
5.3	Réglage dans le domaine fréquentiel	67
5.3.1	Réglage des régulateurs proportionnels	68
5.3.2	Réglage des régulateurs proportionnels-intégrales	69
5.3.3	Réglage des régulateurs proportionnels-dérivés (avance de phase)	72
5.4	Conclusion	76
6	Applications industrielles	77
6.1	Introduction	77
6.2	Régulation de température	77
6.3	Régulation de débit	81
6.4	Régulation de pression	83
6.5	Régulation de niveau	85
6.6	Conclusion	87

Avant-propos

Ce cours a été créé comme un support pédagogique pour les étudiants qui s'intéressent à la régulation industrielle. Il a été élaboré dans le but de fournir une introduction complète et accessible à cette discipline complexe et en constante évolution. Les auteurs ont mis en place un travail rigoureux de recherche, de lecture et de synthèse de diverses sources documentaires, telles que des livres, des cours et des sites internet pour garantir la qualité et l'exactitude des informations présentées.

Il est important de noter que ce cours ne prétend pas couvrir l'ensemble des connaissances en matière de régulation industrielle, mais plutôt de présenter certaines de ses multiples facettes. Les auteurs ont pris soin de sélectionner les concepts clés et les exemples pratiques les plus pertinents pour aider les étudiants à comprendre les fondamentaux de cette discipline.

Enfin, les auteurs tiennent à souligner que la plupart des illustrations et figures présentées dans ce cours sont soumises à des droits d'auteur. Leur utilisation dans ce cours est strictement à des fins pédagogiques et les auteurs ont cité toutes les sources d'origine. Ils encouragent les étudiants à poursuivre leur propre recherche et à explorer davantage les concepts présentés dans ce cours afin de développer une compréhension plus complète et approfondie de la régulation industrielle.

Introduction à la régulation industrielle

1.1 Introduction

Un processus est une action ou une séquence d'actions réalisée dans un but particulier ; Il peut être caractérisé comme un ensemble de procédures modifiant un produit physiquement ou chimiquement. Les opérations industrielles comprennent la fabrication du papier, du ciment, du verre et du traitement de l'eau.

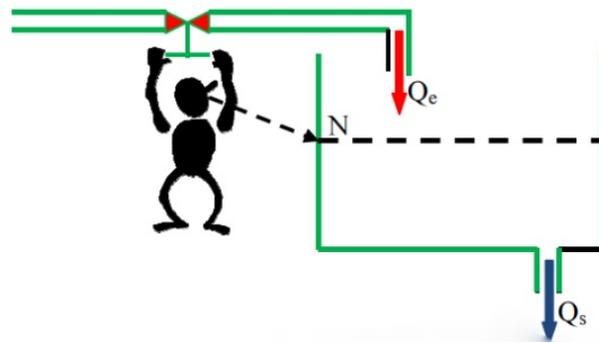
La régulation automatique est une méthodologie des outils et procédures utilisées pour contrôler une ou plusieurs quantités physiques d'un processus (vitesse, température, pression, courant, etc.) afin d'appliquer son comportement prédéterminé. Afin de décider de la ligne de conduite appropriée à effectuer sur le système pour s'assurer qu'elle se comporte comme prévue, les quantités physiques ou les signaux, doivent être mesurées afin de confirmer leur état. Avec la qualification automatique, il est reconnu qu'aucune intervention manuelle n'est requise ; L'opération doit s'exécuter automatiquement sans l'implication d'un opérateur humain. Les dispositifs mécaniques et électroniques sont inévitablement impliqués dans un tel projet.

Malgré les effets de l'environnement, on veut qu'une grandeur physique particulière (vitesse, courant ou température) ait une certaine valeur moyenne en régime permanent (perturbations). Par conséquent, les systèmes de contrôle automatique ont la capacité de modifier le comportement statique et dynamique d'une ou plusieurs grandeurs physiques d'un système, les faisant évoluer en fonction des exigences de l'application.

1.2 Définition

La régulation et le contrôle de processus sont étroitement liés car la régulation consiste à maintenir une grandeur physique à une valeur cible en utilisant un système de contrôle qui prend en compte les perturbations. Les exemples que vous avez donnés illustrent bien cette idée : dans chaque cas, il y a une grandeur physique qui doit être maintenue à une valeur souhaitée, et un système de contrôle est utilisé pour surveiller et ajuster cette grandeur en réponse aux perturbations.

1.3 Analyse des opérations d'une chaîne régulation



Concernant la description du contrôle de niveau d'un réservoir, elle est correcte. En effet, l'opérateur utilise ses sens (vue, ouïe, toucher) pour jauger le niveau, puis compare cette valeur à la valeur de consigne pour déterminer l'action corrective à effectuer (ici, ouvrir ou fermer la vanne). Le cycle "Mesurer, comparer et corriger" est en effet un cycle en boucle fermée, car la correction apportée peut avoir un impact sur la mesure initiale.

Le schéma fonctionnel de la boucle de contrôle que vous avez évoqué est également pertinent. Il s'agit du schéma classique d'une boucle de contrôle en régulation industrielle, composée d'un capteur (qui mesure la grandeur à réguler), d'un comparateur (qui compare la mesure à la consigne) et d'un actionneur (qui agit sur le processus pour corriger l'écart entre la mesure et la consigne). Ce schéma peut être adapté à différents processus industriels, tels que la régulation de la température, de la pression, du débit, etc.

Une personne peut contrôler le niveau d'un réservoir en procédant comme suit : en observant le niveau, en le comparant à la valeur souhaitée, en utilisant la raison pour déterminer l'action à entreprendre et en manipulant la vanne en l'ouvrant légèrement.

Chacune de ces procédures affecte les autres car elles sont effectuées en permanence. A titre d'illustration, si le niveau a tendance à monter, l'individu peut choisir de fermer légèrement la vanne, ce qui réduit le débit et abaisse le niveau.

Afin de maintenir le niveau constant malgré les variations de pression du réseau de distribution d'eau et du débit de sortie, la personne surveille le niveau du réservoir et contrôle le débit. Un humain utilise trois organes pour maintenir le niveau :

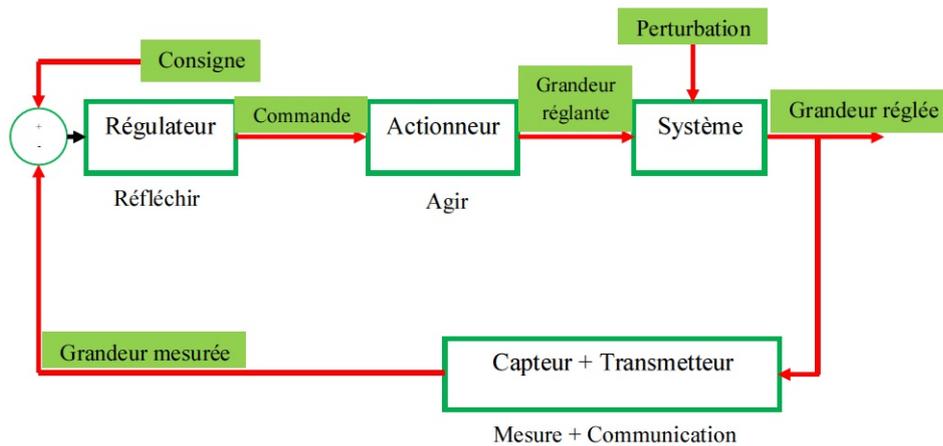
- l'utilisation de l'œil pour jauger le niveau
- la capacité de l'esprit à penser de manière critique et à comparer ;

Par conséquent, la régulation des niveaux implique essentiellement :

- Mesurer par l'œil,
- Comparer/Traiter par le cerveau,
- et Corriger par la main (et la vanne).

Le cycle "Mesurer, comparer et corriger" est un cycle en boucle fermée.

Dans la figure suivante, nous voyons le schéma fonctionnel de la boucle de contrôle.



1.4 Qu'est-ce qui constitue une boucle de contrôle ?

Une boucle de contrôle est un système qui utilise des capteurs pour mesurer une grandeur physique et compare cette grandeur à une valeur de consigne. Ensuite, il utilise un algorithme de contrôle pour déterminer la sortie appropriée pour maintenir la grandeur physique à la valeur de consigne. Cette sortie est appliquée à un actionneur pour modifier la

grandeur physique. La boucle de contrôle est souvent représentée par un schéma fonctionnel qui montre les composants de la boucle, tels que le capteur, l'élément de comparaison, l'élément de commande et l'actionneur.

L'utilisation de technologies pour automatiser les processus de régulation peut aider à éliminer les erreurs humaines et améliorer l'efficacité et la fiabilité de la régulation.

Dans la figure suivante, la boucle de régulation est représentée par un schéma fonctionnel similaire à celui d'une boucle de contrôle. Les capteurs mesurent la grandeur physique (niveau du réservoir), qui est comparée à la valeur de consigne (niveau souhaité) dans l'élément de comparaison. L'élément de commande calcule la sortie appropriée (angle de la vanne) en fonction de l'erreur de régulation (différence entre la grandeur physique mesurée et la valeur de consigne), et l'actionneur (vanne) modifie la grandeur physique (débit) en réponse à la sortie de l'élément de commande. Ce processus se répète en boucle fermée pour maintenir le niveau du réservoir à la valeur de consigne.

D'après la figure suivante la boucle de régulation devient alors :

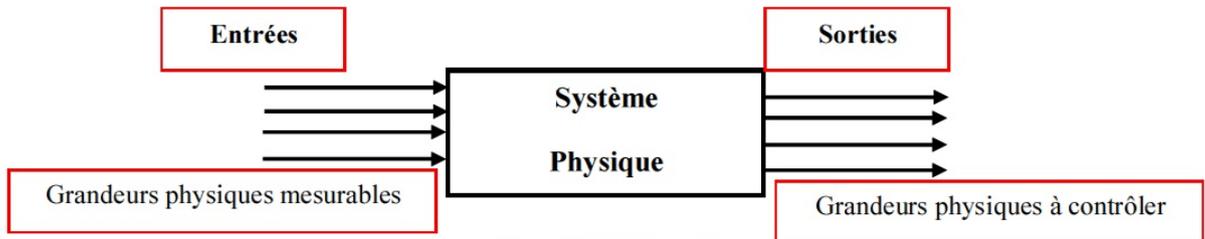
- Le procédé
- La variable manipulée ou grandeur réglante
- La variable commandée ou la grandeur réglée
- La consigne
- Les grandeurs perturbatrices
- Le capteur
- Le régulateur
- L'élément final de commande
- La charge

1.5 Boucle ouverte et boucle fermé

1.5.1 Boucle ouverte

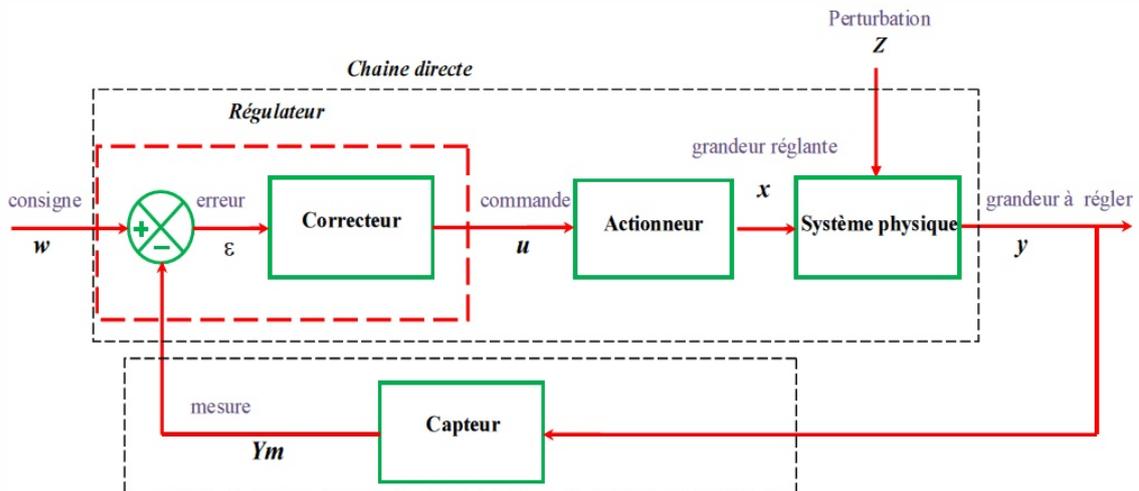
En mode manuel, le régulateur ne reçoit pas de retour d'information de la grandeur à réguler, ce qui signifie que la grandeur de correction n'est pas liée à la grandeur de mesure en temps réel.

Dans ce cas, la boucle est ouverte et la précision de la régulation dépend entièrement de l'expertise et de l'expérience de l'opérateur. Ce mode est souvent utilisé pour des tâches de contrôle simples et ne nécessitant pas une grande précision.



1.5.2 Boucle fermé

Dans une boucle fermée, la grandeur mesurée est comparée à la valeur de consigne et l'écart est utilisé pour calculer une commande de correction qui est envoyée à l'actionneur. La grandeur de la correction est donc dépendante de la grandeur de la mesure, ce qui permet de maintenir la grandeur mesurée à sa valeur de consigne malgré les perturbations. La boucle fermée est également appelée boucle de rétroaction, car elle utilise l'information de la mesure pour ajuster la commande.



1.6 Exemples de système de régulation (Véhicule électrique)

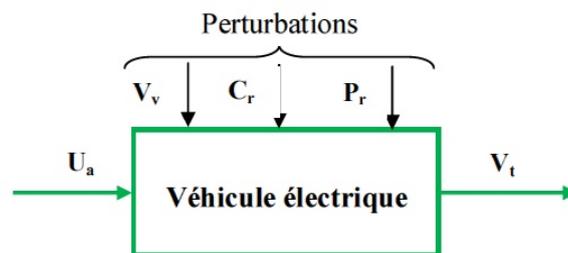
En général, les sorties des systèmes dépendent de plus que de simples valeurs quantifiables ; ils sont également sensibles à des grandeurs supplémentaires, appelées perturbations, qui varient de manière imprévisible. A titre illustratif, la fluctuation de la tension d'induit U_a (grandeur mesurable) des moteurs affecte la variation de la vitesse V_t d'une voiture électrique. Cependant, la vitesse du véhicule est également influencée par des facteurs tels que la charge C_r , la pente de la route P_r , la vitesse frontale du vent V_v et sa direction.

La régulation des véhicules électriques est un domaine en pleine expansion en raison de la demande croissante de mobilité durable. Les systèmes de régulation sont utilisés pour contrôler la vitesse, le couple et la puissance des moteurs électriques, ainsi que pour optimiser l'efficacité énergétique et la gestion de la batterie.

Dans les véhicules électriques, la régulation est souvent réalisée à l'aide de boucles de rétroaction fermées qui mesurent la vitesse du moteur, le courant, la tension de la batterie et d'autres grandeurs pour ajuster les paramètres du système en temps réel. Les perturbations telles que les variations de la tension de la batterie, les changements de charge et les perturbations environnementales sont également prises en compte pour assurer une régulation précise et fiable.

En résumé, la régulation des véhicules électriques est un exemple important de l'utilisation de systèmes de régulation pour contrôler des systèmes complexes en temps réel, afin d'optimiser leur performance et leur efficacité énergétique.

Ce système peut être illustré par la figure suivante, qui représente des grandeurs non mesurables :



1.7 Conclusion

En conclusion, la régulation industrielle est un domaine clé pour assurer la stabilité et l'efficacité des processus industriels. Elle permet de maintenir des variables telles que la température, la pression, le débit, le niveau, etc. dans des plages de valeurs désirées, tout en prenant en compte les perturbations et les variations des conditions de fonctionnement.

Les systèmes de régulation peuvent être de différents types et présentent des caractéristiques complexes qui nécessitent des approches de conception et de contrôle spécifiques. Les régulateurs PID sont les plus couramment utilisés, mais d'autres régulateurs tels que les régulateurs tout ou rien et les régulateurs flous peuvent être plus adaptés à certains types de processus.

La régulation industrielle est un domaine en constante évolution, avec de nouveaux défis liés à l'optimisation énergétique, à la sécurité, à l'automatisation et à l'intégration des systèmes. Les progrès de la technologie de l'information et de la communication offrent également de nouvelles opportunités pour améliorer la performance des systèmes de régulation industrielle et les rendre plus flexibles, intelligents et interconnectés.

Régulateur tout ou rien

2.1 Introduction

Outre le phénomène de saturation, certains systèmes ou composants d'un système présentent un comportement non linéaire du fait de leur construction, de contraintes technologiques ou, plus simplement, de leur propre principe de fonctionnement.

2.2 Systèmes tout ou rien

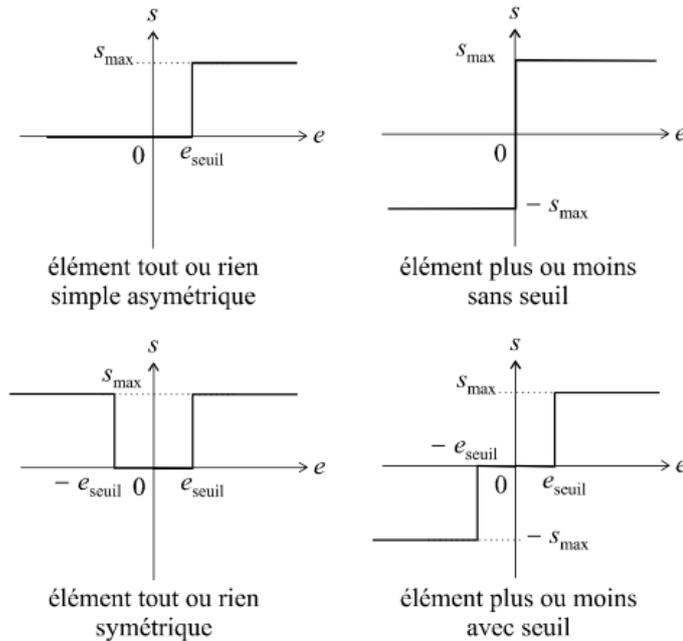
Les systèmes tout ou rien sont des systèmes dont la sortie est binaire, c'est-à-dire qu'elle ne peut prendre que deux valeurs possibles, souvent 0 ou 1, vrai ou faux, ou encore ouvert ou fermé. La valeur de la sortie dépend de la plage dans laquelle se situe la valeur de l'entrée, qui peut être un signal électrique, une pression, une température, etc.

Dans les systèmes tout ou rien, la caractéristique d'entrée-sortie peut être représentée sous la forme d'une courbe en escalier, avec des sauts brusques entre les deux valeurs de sortie possibles. La forme de cette courbe dépend de la nature du système et de la manière dont il est conçu.

Certains systèmes tout ou rien peuvent être appelés "plus ou moins", car la sortie peut prendre une troisième valeur intermédiaire, telle que 0,5. Ces systèmes sont souvent utilisés dans les systèmes de régulation pour ajuster la quantité de ressources utilisées en fonction des besoins, tout en évitant une utilisation excessive.

D'autres systèmes tout ou rien peuvent avoir une caractéristique avec un seuil, ce qui signifie que la sortie ne change pas tant que la valeur d'entrée est inférieure ou supérieure à un certain seuil. Par exemple, un capteur de température peut déclencher une alarme lorsque la température dépasse un certain seuil.

Enfin, certains systèmes tout ou rien peuvent avoir une caractéristique sans seuil, ce qui signifie que la sortie change instantanément dès que la valeur d'entrée franchit une certaine valeur, sans palier intermédiaire. Ces systèmes peuvent être utilisés dans les systèmes de sécurité, tels que les détecteurs d'incendie, où une réponse rapide est essentielle.



Les relais électriques, qui sont des organes de commande fréquemment utilisés, possèdent des caractéristiques de ce type.

* Exemple :

L'entraînement électrique en position est décrit par l'équation suivante sans couple résistant parasite :

$$\ddot{\theta}(t) + \frac{1}{J} \left(f + \frac{(K\phi)^2}{R} \right) \dot{\theta}(t) = \frac{K\phi}{R} u(t)$$

avec des valeurs numériques données :

$$\ddot{\theta}(t) + 2\dot{\theta}(t) = 4u(t)$$

La fonction de transfert équivalente est :

$$\Theta(p) = \frac{4}{(p+2)}U(s)$$

soit, en revenant dans le domaine du temps et en introduisant l'écart de position $\varepsilon = \theta_c - \theta$ où θ_c est constante.

$$\dot{\varepsilon}(t) = -\dot{\theta}(t)$$

$$\ddot{\varepsilon}(t) = -\ddot{\theta}(t)$$

Par substitution dans l'équation différentielle précédente, on trouve :

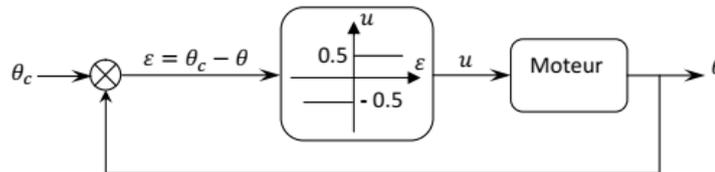
$$\ddot{\varepsilon}(t) + 2\dot{\varepsilon}(t) = -4u\varepsilon(t)$$

Un régulateur tout ou rien est utilisé :

$$u(\varepsilon) = \begin{cases} 0.5 & \text{si } \varepsilon > 0 \\ -0.5 & \text{si } \varepsilon \leq 0 \end{cases}$$

Les deux équations différentielles peuvent être résolues analytiquement, mais actuellement, il est plus pratique de résoudre numériquement ce genre de problème.

Les trajectoires dans le plan de phase $(\varepsilon, \dot{\varepsilon})$, où l'écart est porté en abscisse et la dérivée $\dot{\varepsilon}$ de l'écart ε en ordonnée, sont obtenues en résolvant ces deux équations différentielles. Ainsi, nous arrivons au schéma fonctionnel ci-dessous :



Initialement, la position $\theta(0)$ et la vitesse $\dot{\theta}(0)$ sont toutes deux nulles, de sorte que :

$$\varepsilon(0) = \theta_c - \theta(0) = \theta_c$$

$$\dot{\varepsilon}(t) = -\dot{\theta}(t) = 0$$

Le signal d'entrée est l'écart ε entre la consigne θ_c et la position actuelle θ du moteur. Ce signal est traité par un régulateur tout ou rien, qui génère un signal de commande binaire (0 ou 1) pour le moteur électrique.

Le signal de commande binaire est ensuite appliqué au moteur électrique, qui produit une vitesse angulaire $\dot{\theta}$ proportionnelle au signal de commande. La vitesse angulaire est ensuite convertie en position angulaire θ par intégration.

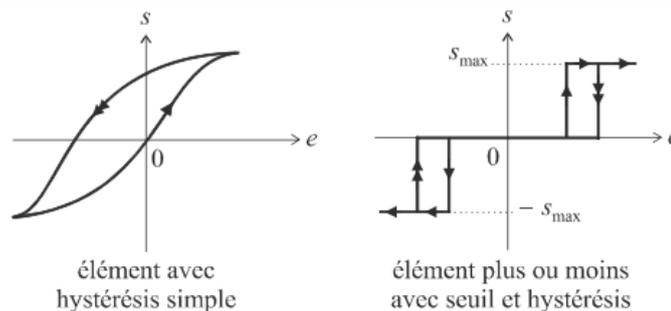
Les trajectoires dans le plan de phase $(\varepsilon, \dot{\varepsilon})$ peuvent être tracées en résolvant numériquement les équations différentielles décrivant le système. Les trajectoires représentent l'évolution de l'écart et de sa dérivée au fil du temps pour différentes conditions initiales. Ces trajectoires permettent de visualiser le comportement du système et d'analyser sa stabilité.

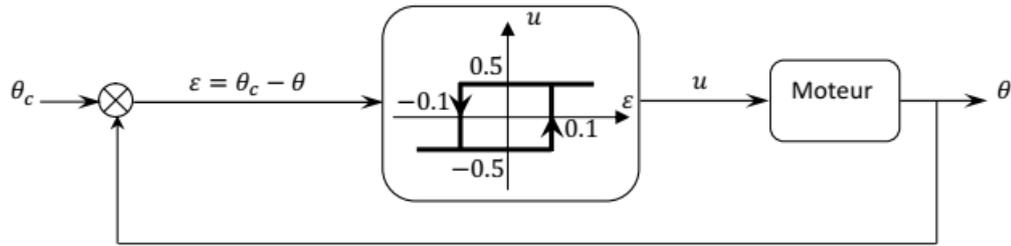
2.3 Systèmes à hystérésis

Les systèmes à hystérésis sont des systèmes non linéaires qui présentent une mémoire, c'est-à-dire que leur réponse dépend de l'historique des signaux d'entrée plutôt que seulement de l'instant présent.

Les courbes d'hystérésis peuvent être trouvées dans de nombreux systèmes physiques et électroniques, tels que les matériaux magnétiques, les circuits électroniques, les systèmes hydrauliques et pneumatiques, etc. Dans les matériaux magnétiques, par exemple, l'hystérésis est la courbe de réponse du matériau à un champ magnétique, qui est différente selon que le champ magnétique augmente ou diminue.

Les systèmes à hystérésis peuvent être modélisés mathématiquement de différentes manières, mais la modélisation la plus courante utilise des équations différentielles non linéaires qui décrivent la dynamique de la réponse du système en fonction de l'historique des signaux d'entrée. La modélisation des systèmes à hystérésis peut être complexe en raison de leur non-linéarité et de leur mémoire, mais elle est souvent nécessaire pour comprendre et contrôler le comportement de ces systèmes dans de nombreuses applications.





Les organes présentant des non linéarités qui se traduisent par la présence, en sortie, de quelques valeurs discrètes (éléments tout ou rien ou plus ou moins) sont encore appelées non linéarités de type relais.

* Exemple :

Soit à nouveau l'entraînement en position des exemples précédents, en utilisant maintenant un régulateur tout ou rien avec hystérésis.

La figure suivante montre des trajectoires dans le plan de phase.

Il existe deux demi-droites de commutations, à savoir :

$$\varepsilon = \begin{cases} 0.1 & \text{si } \dot{\varepsilon} > 0 \\ -0.1 & \text{si } \dot{\varepsilon} < 0 \end{cases}$$

Dans cet exemple, un régulateur tout ou rien avec hystérésis est utilisé pour contrôler la position d'un objet en utilisant des actionneurs tout ou rien. La figure montre les trajectoires dans le plan de phase du système contrôlé, avec deux demi-droites de commutation définies pour l'hystérésis.

Les demi-droites de commutation indiquent quand le régulateur doit changer d'état (passer de "on" à "off" ou vice versa). Si la dérivée de l'erreur de position est positive et supérieure à 0.1, le régulateur basculera l'actionneur en mode "off". Si la dérivée de l'erreur de position est négative et inférieure à -0.1 , le régulateur basculera l'actionneur en mode "on". Ces valeurs d'hystérésis sont utilisées pour éviter de commuter rapidement entre les deux états en raison des petites variations autour de la position désirée.

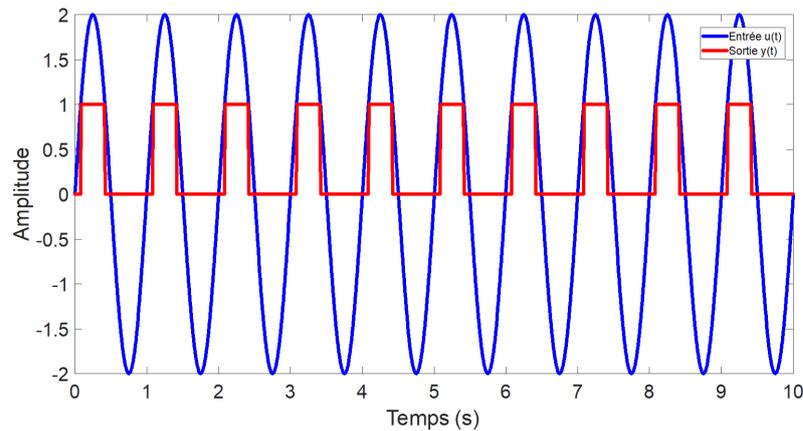
Le cycle limite, qui prend la forme d'une trajectoire proche d'elle-même, est un comportement périodique du système contrôlé. L'imprécision de la consigne de position permet de limiter la fréquence des commutations du régulateur tout ou rien avec hystérésis au fil

du temps. La figure de $u(t)$ pour une consigne de position de 1 montre cette limitation de la fréquence de commutation par rapport à un régulateur tout ou rien de base.

Particulièrement évident est le cycle limite, qui prend la forme d'une trajectoire proche d'elle-même.

Il est clair que la consigne n'est pas exactement affectée à la quantité à réguler. Comme le montre la figure de $u(t)$ pour $\theta_c = 1$, cette imprécision permet de limiter, contrairement au marche/arrêt de base du régulateur, la fréquence des commutations au fil du temps.

Le cycle limite, sous la forme d'une trajectoire fermée sur elle-même, est particulièrement visible. La consigne n'est évidemment pas exactement assignée à la grandeur à commander. Cette imprécision permet de limiter, contrairement au régulateur tout ou rien simple, la fréquence des commutations quand le temps s'écoule, comme il ressort de la figure de $u(t)$ pour $\theta_c = 1$.



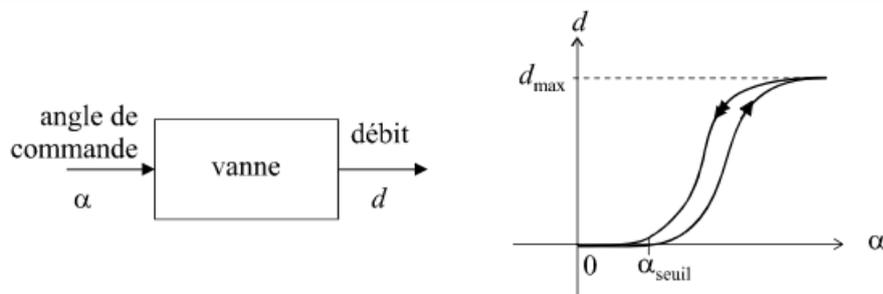
2.4 Caractéristiques complexes

La vanne hydraulique est un exemple d'appareil présentant des fonctionnalités complexes avec des phénomènes de seuil, de saturation et d'hystérésis. Le signal d'entrée de la vanne est l'angle d'ouverture et le signal de sortie est le débit du fluide qu'elle laisse passer.

Le phénomène de seuil se produit lorsque l'angle est trop faible pour que la vanne réponde et que le fluide ne commence à s'écouler qu'à partir d'un certain angle donné. Cependant, la relation entre l'angle d'ouverture et le débit n'est pas toujours proportionnelle car la caractéristique de la vanne n'est pas une ligne droite.

Le débit augmente à mesure que l'angle augmente, mais après une certaine valeur d'angle, le débit est à son maximum même si nous continuons à tourner la vanne. Cela correspond au phénomène de saturation. Si l'on diminue l'angle à nouveau, le jeu mécanique peut causer un phénomène d'hystérésis, où le débit ne recommence à décroître que lorsque l'on a « rattrapé » le jeu mécanique.

Ces phénomènes de seuil, de saturation et d'hystérésis peuvent rendre le contrôle de la vanne plus difficile, car la relation entre l'angle d'ouverture et le débit est complexe et non linéaire. Pour cette raison, les ingénieurs doivent souvent utiliser des régulateurs spéciaux pour contrôler le débit de la vanne en fonction de la position de l'angle d'ouverture, en tenant compte de ces phénomènes pour obtenir un comportement désiré et stable.



2.5 Conclusion

En conclusion, le régulateur tout ou rien est un type de régulateur utilisé dans les systèmes de contrôle automatique pour lesquels la sortie ne peut prendre que deux valeurs possibles. Ce type de régulateur est souvent utilisé lorsque le système est très non linéaire, ou lorsqu'il est difficile ou coûteux d'utiliser des régulateurs plus sophistiqués.

Le régulateur tout ou rien peut être utilisé pour des systèmes électromécaniques tels que l'entraînement électrique en position. Dans ce cas, le régulateur tout ou rien permet de contrôler la position de l'entraînement en ajustant la tension d'entrée en fonction de la valeur de l'écart de position mesurée. Les trajectoires dans le plan de phase $(\varepsilon, \dot{\varepsilon})$ sont utilisées pour visualiser l'évolution du système et pour évaluer sa stabilité.

Bien que le régulateur tout ou rien soit relativement simple et peu coûteux à mettre en œuvre, il présente également des inconvénients tels que des transitions brutales entre les états et des risques d'oscillations ou de régulation incorrecte en raison de la non-linéarité des systèmes. Il est donc important de choisir le type de régulateur approprié en fonction des caractéristiques du système et des objectifs de régulation.

Identification des systèmes en boucle ouverte et fermée

3.1 Introduction

Dans un système en boucle ouverte, la sortie du système n'affecte pas l'entrée. En d'autres termes, le signal de sortie n'est pas régulé ou corrigé en fonction de la mesure de la sortie. Les systèmes en boucle ouverte sont généralement utilisés lorsque la sortie n'a pas besoin d'être régulée ou lorsque la sortie est utilisée pour une autre application.

En revanche, dans un système en boucle fermée, la sortie est mesurée et comparée à une valeur de référence. Si la mesure diffère de la valeur de référence, un signal de correction est généré pour réduire l'écart. Les systèmes en boucle fermée sont utilisés lorsque la précision et la stabilité sont nécessaires.

L'identification des systèmes en boucle ouverte et fermée est importante pour déterminer les méthodes de contrôle appropriées et pour améliorer la performance du système de régulation.

3.2 But de l'identification

Il est nécessaire d'évaluer le comportement statique et dynamique de ce procédé seul ou instrumenté, c'est-à-dire de comprendre sa fonction de transfert régulateur, afin d'atteindre les objectifs définis dans le cahier des charges de régulation d'un procédé.

Une équation du système (un système simple ou un système qui peut être divisé en éléments simples) peut être utilisée pour déterminer la fonction de transfert, qui se traduit par un modèle de connaissances. Il est préférable d'utiliser l'identification expérimentale

lorsque le processus est compliqué et que son équation est difficile, voire impossible. L'examen des réponses temporelles ou fréquentes fournies directement au système soumis à un certain signal d'entrée constitue la base d'une telle identification.

3.3 Choix du modèle

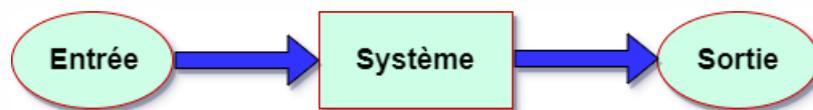
Il est essentiel de rechercher un modèle mathématique d'un processus, et ce modèle doit refléter fidèlement le comportement du processus. Le modèle doit cependant trouver un équilibre entre ne pas être trop simple pour ne pas masquer certains éléments nuisibles au bon fonctionnement et ne pas être trop complexe pour ne pas être incompatible avec le correcteur désormais accessible.

Test en chaîne ouverte (le système recherché n'est pas asservi) et test en chaîne fermée sont deux techniques d'identification à prendre en considération (un régulateur asservit le système).

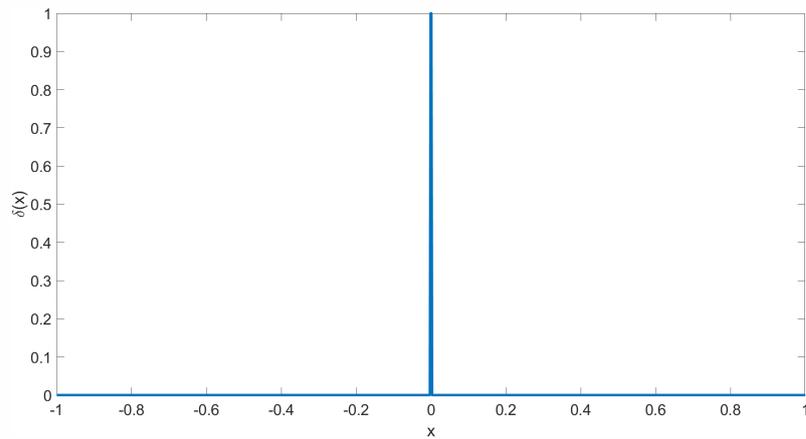
3.4 Identification en chaîne ouverte

Le processus qui doit être identifié comme n'est plus automatiquement examiné lors d'un tel test. Pour pouvoir réagir au signal de commande, le régulateur est commuté en mode manuel. Pour s'assurer que le système est linéaire et que le produit déjà en cours n'est pas perturbé, l'automaticien ne doit introduire que des fluctuations mineures (quelques %) autour d'un point de fonctionnement choisi. Ils doivent également s'assurer que la perturbation n'affectera pas cette identification.

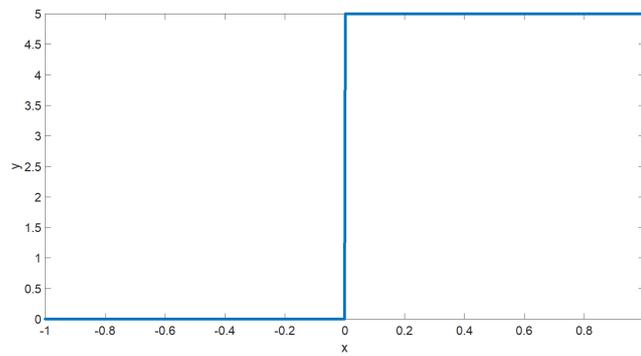
Nous enregistrons le signal de sortie $x(t)$ puis l'évaluons après avoir envoyé un signal d'entrée connu $y(t)$ (impulsion, échelon ou rampe).



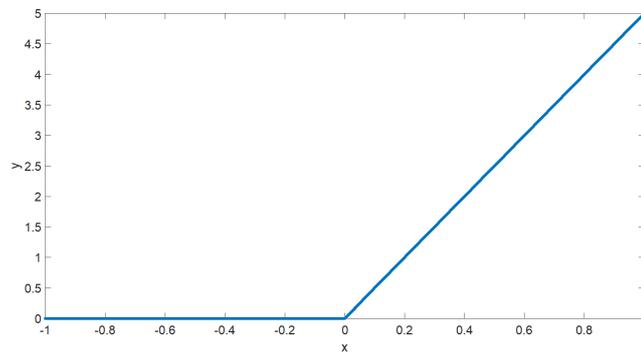
Les deux signaux les plus employés sont l'échelon et la rampe. L'échelon est le plus facile et rapide à réaliser mais peut provoquer des variations assez brusques sur des procédés très sensibles. On utilise la rampe pour que le procédé subisse une variation plus douce et donc plus acceptable au niveau sécurité du produit et de l'installation.



Impulsion unitaire



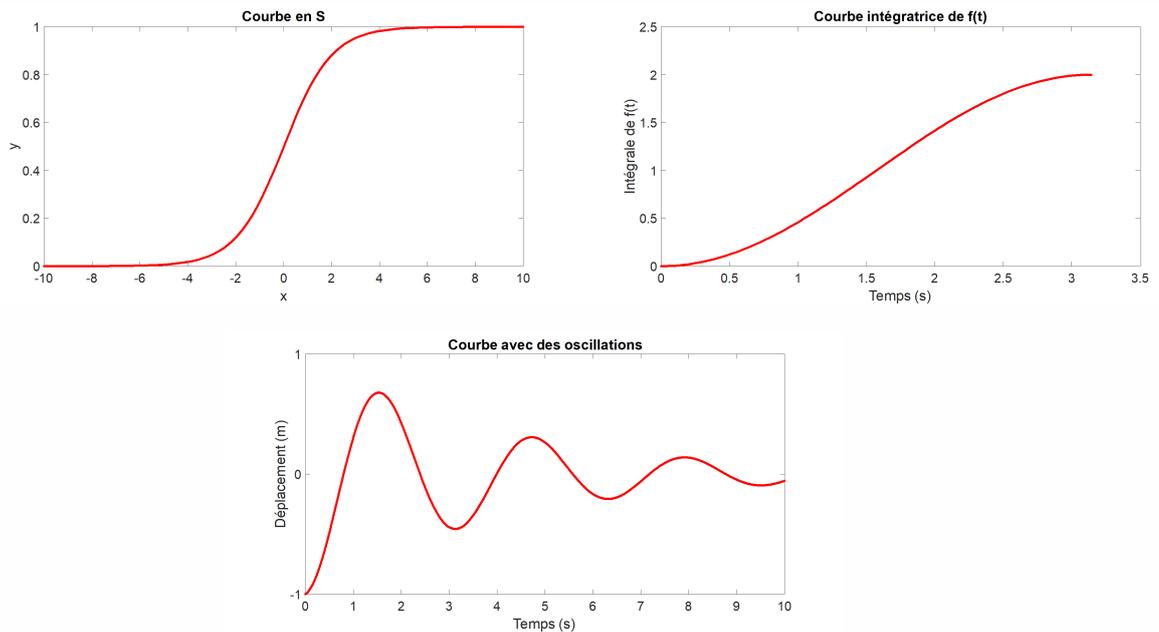
Échelon d'amplitude $A=5$



Rampe de pente 45°

On enregistre le signal de sortie. Lorsque le signal d'entrée est une impulsion, la courbe obtenue est appelée réponse impulsionnelle. Pour un signal d'entrée en échelon, la courbe obtenue est appelée réponse indicielle (ou réponse à un échelon de position). La courbe obtenue à une rampe est dite réponse à un échelon de vitesse.

Les trois grandes familles de courbes les plus rencontrées sont les suivantes :



La forme de la courbe résultante offre un mécanisme d'application pour simuler le processus. Il est simple d'identifier le modèle pour les courbes facilement reconnaissables, telles que la réponse indicielle du premier ordre.

Si le système n'a pas de dépassement et que sa pente présente une discontinuité à l'origine, on peut choisir de le modéliser par un premier ordre de la forme : $\frac{K}{1+\tau p}$.

Les paramètres K et τ peuvent être identifiés comme suit :

Obtenez les données de réponse du système à une entrée connue, par exemple une réponse de type échelon unitaire.

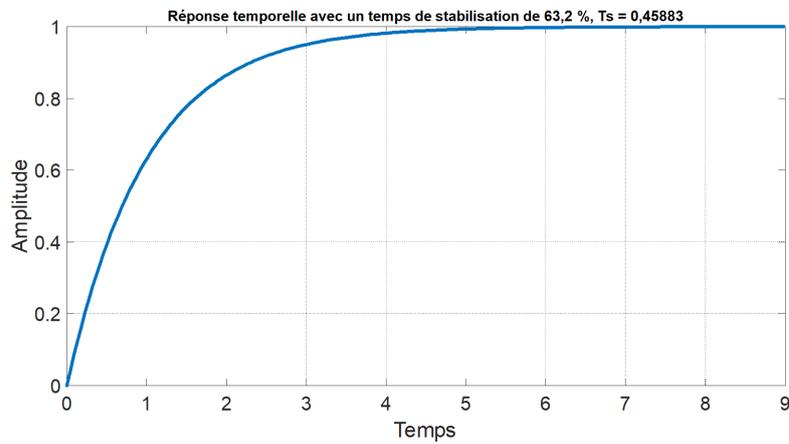
Identifiez le temps de montée (t_r) de la réponse, qui est le temps nécessaire pour que la réponse atteigne 63,2% de sa valeur finale. Cela correspond à un temps de montée de $\tau * \ln(2)$ pour un système du premier ordre.

Identifiez la valeur finale (y_f) de la réponse, qui est la valeur que la réponse atteint une fois qu'elle a convergé.

Calculez le gain K comme la différence entre la valeur finale et la valeur initiale de la réponse, divisée par l'amplitude de l'entrée. Si l'entrée est un échelon unitaire, alors K est simplement égal à y_f .

Calculez la constante de temps τ comme le temps de montée divisé par le logarithme naturel de 2, c'est-à-dire $\tau = t_r / \ln(2)$.

En utilisant ces étapes, vous pouvez identifier les paramètres K et τ pour un système du premier ordre à partir de données de réponse expérimentales ou simulées. Cependant, il convient de noter que cette méthode peut ne pas être précise si le système est non-linéaire ou si la réponse contient des oscillations ou des retards. Dans de tels cas, des méthodes plus avancées peuvent être nécessaires pour identifier les paramètres du système.



La réponse indicielle d'un système du premier ordre est donnée par :

$$y(t) = K(1 - e^{-t/\tau})$$

où $y(t)$ est la sortie du système à l'instant t , K est le gain statique et τ est la constante de temps du système.

En utilisant les valeurs données dans la question, nous avons :

$$K = 1 \text{ et } \tau = 1$$

En substituant ces valeurs dans l'expression de la réponse indicielle, nous avons :

$$y(t) = 1 - e^{-t}$$

Nous pouvons maintenant trouver la fonction de transfert en utilisant la transformée de Laplace :

$$Y(p) = \frac{1}{p} - \frac{1}{(p+1)}$$

$$\frac{Y(p)}{U(p)} = G(p) = \frac{1}{p(p+1)}$$

où $U(p)$ est la transformée de Laplace de l'entrée et $Y(p)$ est la transformée de Laplace de la sortie.

Ainsi, la fonction de transfert du système du premier ordre correspondant à la réponse indicielle donnée est :

$$G(p) = \frac{1}{p(p+1)}$$

3.4.1 Courbe en S

3.4.1.1 Méthode de Strejc

Le modèle de Strejc est un modèle mathématique utilisé pour décrire la dynamique des systèmes thermiques. Dans ce modèle, la dynamique du système est représentée par j constantes de temps τ identiques.

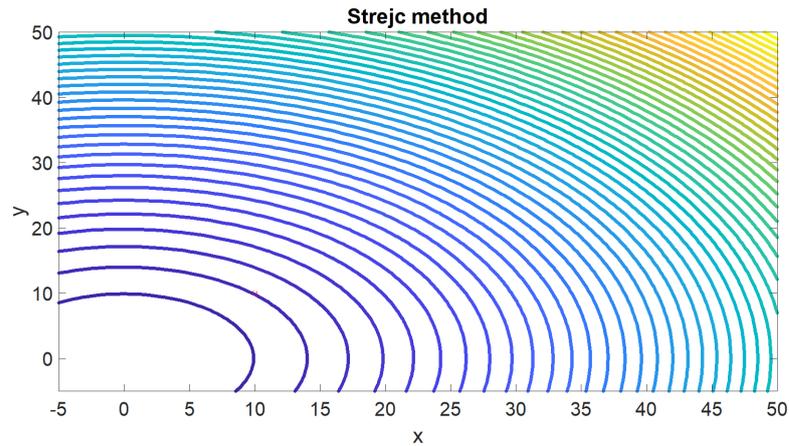
La fonction de transfert du modèle de Strejc est donnée par :

$$H(p) = \frac{K}{(1 + \tau p)^n}$$

où K est le gain statique du système, p est la variable de Laplace, τ est la constante de temps du système et n est le nombre de constantes de temps identiques.

L'utilisation de multiples constantes de temps identiques dans le modèle de Strejc permet de modéliser des systèmes thermiques qui présentent une dynamique similaire dans différentes parties du système. Cela permet de simplifier le modèle mathématique tout en conservant une bonne précision pour décrire la dynamique globale du système.

Cependant, il est important de noter que la dynamique réelle des systèmes thermiques peut être plus complexe que ce que le modèle de Strejc peut représenter, et que des modèles plus sophistiqués peuvent être nécessaires pour décrire avec précision le comportement de ces systèmes.



Le modèle de Strejc pour un système intégrateur est une représentation mathématique d'un système physique qui peut être utilisé pour analyser son comportement. Ce modèle est souvent utilisé en ingénierie, en contrôle de processus et en traitement du signal pour modéliser des systèmes qui intègrent l'entrée au fil du temps.

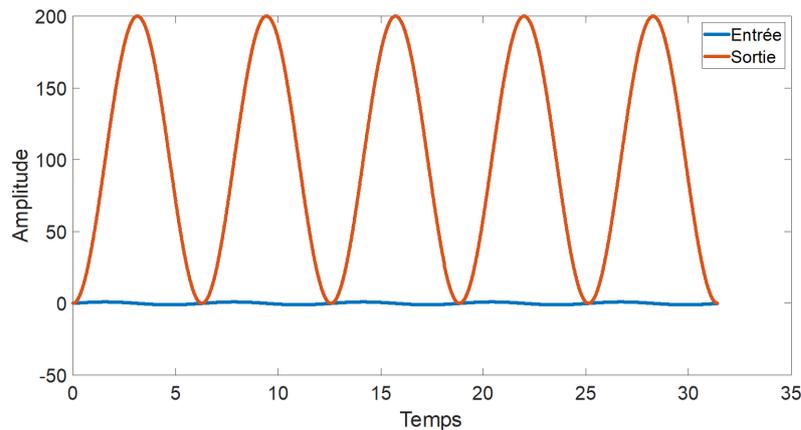
Le modèle de Strejc est généralement représenté sous la forme d'une équation différentielle linéaire du premier ordre de la forme :

$$y(t) = K \int [x(\tau)d\tau] + y(0)$$

où $y(t)$ est la sortie du système à l'instant t , $x(t)$ est l'entrée du système à l'instant t , $y(0)$ est la valeur initiale de la sortie à l'instant $t = 0$, K est une constante de proportionnalité et $\int [x(\tau)d\tau]$ représente l'intégration de l'entrée sur l'intervalle de temps $[0, t]$.

Ce modèle peut être utilisé pour analyser le comportement d'un système intégrateur et pour déterminer sa réponse à différentes entrées. Par exemple, si l'entrée est une fonction échelon unitaire, le modèle de Strejc prédit que la sortie du système sera une fonction rampe linéaire.

Le modèle de Strejc est un modèle simplifié qui ne tient pas compte de certains effets physiques tels que la saturation, le bruit et les retards. Cependant, il est souvent utilisé comme point de départ pour développer des modèles plus complexes et plus précis de systèmes intégrateurs.



3.4.1.2 Méthode de Strejc-Davoust

La méthode de Strejc-Davoust est une méthode de résolution de problèmes qui peut être appliquée à divers domaines, y compris la gestion du temps en milieu professionnel. Le principe de détermination du temps mort est un concept qui peut être utilisé dans le cadre de la méthode de Strejc-Davoust pour optimiser l'utilisation du temps dans une situation de travail donnée.

Le temps mort se réfère au temps où un travailleur n'est pas productif ou n'est pas en train de travailler sur une tâche spécifique. Le principe de détermination du temps mort implique d'analyser les processus de travail pour identifier les sources de temps mort et de développer des solutions pour les réduire ou les éliminer complètement.

Pour appliquer le principe de détermination du temps mort dans le cadre de la méthode de Strejc-Davoust, les étapes suivantes peuvent être suivies :

1. Identifier les activités et les tâches qui sont essentielles pour accomplir un objectif spécifique.
2. Évaluer la durée de chaque activité ou tâche en utilisant des données quantitatives ou des estimations subjectives.
3. Analyser les processus de travail pour identifier les sources de temps mort, y compris les interruptions, les temps d'attente, les déplacements inutiles, etc.
4. Élaborer des solutions pour réduire ou éliminer le temps mort, comme l'optimisation des itinéraires, la réorganisation du lieu de travail, l'utilisation de techniques de gestion du temps, etc.
5. Mettre en place les solutions identifiées et évaluer leur efficacité pour réduire ou éliminer le temps mort.

En appliquant le principe de détermination du temps mort dans le cadre de la méthode de Strejc-Davoust, il est possible d'optimiser l'utilisation du temps et d'améliorer la productivité et l'efficacité dans les environnements professionnels.

3.4.1.3 Méthode de Broïda

La méthode de Broïda utilise une modélisation mathématique simple pour évaluer la qualité des systèmes en fonction de plusieurs critères. Cette modélisation mathématique implique l'utilisation de pondérations et de notes pour chaque critère, ainsi que le calcul de scores totaux pour chaque alternative.

Voici les étapes détaillées pour la modélisation mathématique de la méthode de Broïda :

1. Identifiez les critères de qualité importants pour le système en question. Ces critères peuvent être représentés par des variables notées X_1, X_2, \dots, X_n .
2. Déterminez les pondérations relatives de chaque critère. Ces pondérations sont notées w_1, w_2, \dots, w_n et sont exprimées sous forme de pourcentages ou de poids.
3. Évaluez chaque alternative ou solution en fonction des critères pondérés. Pour chaque alternative, attribuez une note pour chaque critère noté A_{ij} , où i représente le numéro de l'alternative et j représente le numéro du critère.
4. Calculez le score total pour chaque alternative. Pour chaque alternative, calculez le score total en multipliant chaque note de critère par la pondération relative du critère, puis en ajoutant les résultats.
5. Sélectionnez la meilleure alternative. La meilleure alternative est celle qui a le score total le plus élevé.

Voici une formule générale pour calculer le score total pour une alternative i :

$$\text{Score total pour l'alternative } i = \sum_j (w_j \times A_{ij})$$

où w_j est la pondération relative du critère j et A_{ij} est la note de l'alternative i pour le critère j .

La méthode de Broïda est un exemple de modélisation mathématique simple qui peut être utilisée pour résoudre des problèmes multicritères. Cette méthode peut être mise en œuvre dans des feuilles de calcul, des logiciels de modélisation ou des langages de programmation pour faciliter les calculs et la sélection de la meilleure alternative.

- **Exemple**

Supposons qu'il y ait trois critères de qualité, notés X_1 , X_2 et X_3 , et trois alternatives notées A_1 , A_2 et A_3 . Les pondérations relatives pour chaque critère sont 0.4, 0.3 et 0.3, respectivement. Les notes pour chaque alternative pour chaque critère sont données dans le tableau ci-dessous :

Critères	X_1	X_2	X_3
A_1	5	7	9
A_2	8	6	7
A_3	6	9	5

Voici le code MATLAB correspondant :

```
% Données d'entrée w = [0.4, 0.3, 0.3]; % pondérations relatives pour chaque critère A =
[5, 7, 9; 8, 6, 7; 6, 9, 5]; % notes pour chaque alternative pour chaque critère
% Calcul des scores totaux pour chaque alternative score_total = sum(bsxfun(@times, A,
w), 2); % bsxfun réalise une multiplication élément par élément pour chaque ligne de la
matrice A
% Sélection de la meilleure alternative [meilleure_score, meilleure_alternative] =
max(score_total);
```

Le code ci-dessus calcule les scores totaux pour chaque alternative en multipliant chaque note de critère par la pondération relative correspondante, puis en additionnant les résultats. La fonction `bsxfun` est utilisée pour réaliser la multiplication élément par élément pour chaque ligne de la matrice A . La fonction `sum` calcule la somme des résultats de la multiplication, et la fonction `max` est utilisée pour sélectionner la meilleure alternative en fonction du score total le plus élevé.

Le résultat de l'exécution du code ci-dessus est stocké dans les variables `meilleure_score` et `meilleure_alternative`, qui contiennent respectivement le score total le plus élevé et le numéro de l'alternative correspondante.

3.4.2 Courbe intégratrice

Deux méthodes sont utilisables; la première méthode consiste à utiliser une courbe intégratrice pour déterminer le comportement dynamique d'un système. Cette méthode

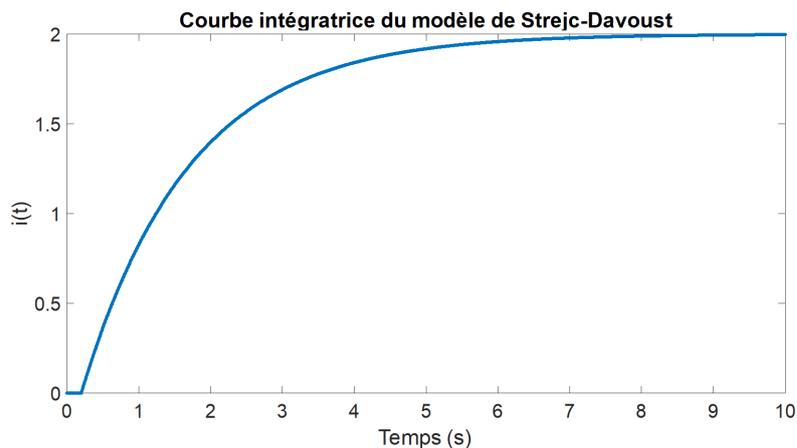
conduit à un modèle mathématique fin, qui prend en compte les effets de la dynamique du système. Cependant, l'analyse de ce modèle peut être délicate, car elle nécessite souvent l'utilisation de techniques mathématiques avancées, telles que la transformée de Laplace ou la théorie du contrôle.

La seconde méthode consiste à utiliser une approche plus grossière, qui consiste à modéliser le système comme étant un système intégrateur. Cette méthode est plus simple et plus rapide à analyser que la première méthode, mais elle ne prend pas en compte tous les effets dynamiques du système. En effet, elle suppose que les variations du signal d'entrée sont suffisamment lentes pour que le système puisse suivre les variations du signal en temps réel.

Il est important de noter que l'utilisation d'une courbe intégratrice ne garantit pas toujours que le système sera effectivement un système intégrateur. En effet, la courbe obtenue dépend du signal appliqué à l'entrée, et si ce signal varie rapidement, le système peut ne pas être en mesure de suivre ces variations en temps réel. Cela peut conduire à des comportements dynamiques non prévus, tels que des oscillations ou des instabilités.

Voici un exemple de code MATLAB pour tracer la courbe intégratrice correspondant au modèle de Strejc-Davoust :

```
K = 2; % gain du système tau = 1.5; % constante de temps d'intégration theta = 0.2; % temps mort
t = 0:0.01:10; % vecteur de temps
i = K * (1 - exp(-(t - theta)/tau)) .* (t >= theta); % courbe intégratrice
plot(t, i); xlabel('Temps (s)'); ylabel('i(t)'); title('Courbe intégratrice du modèle de Strejc-Davoust');
```



Dans cet exemple, la courbe intégratrice est tracée pour un gain $K = 2$, une constante de temps d'intégration $\tau = 1,5$ et un temps mort $\theta = 0,2$. Le vecteur de temps t est défini de 0 à 10 secondes avec un pas de 0,01 seconde. La fonction i calcule la courbe intégratrice en utilisant l'expression de la transformée inverse de Laplace, et le graphique est tracé en utilisant la fonction `plot`.

3.4.3 Courbe avec oscillations

Le modèle proposé pour un système avec une courbe avec oscillations est celui d'un système du second ordre avec retard, autorégulant ou intégrateur.

$$H(p) = \frac{K e^{-Tp}}{1 + 2\zeta \frac{p}{\omega_n} + \frac{p^2}{\omega_n^2}}$$

Ce modèle mathématique décrit la réponse temporelle d'un système à une perturbation en termes d'un oscillateur harmonique amorti, qui est caractérisé par les paramètres suivants :

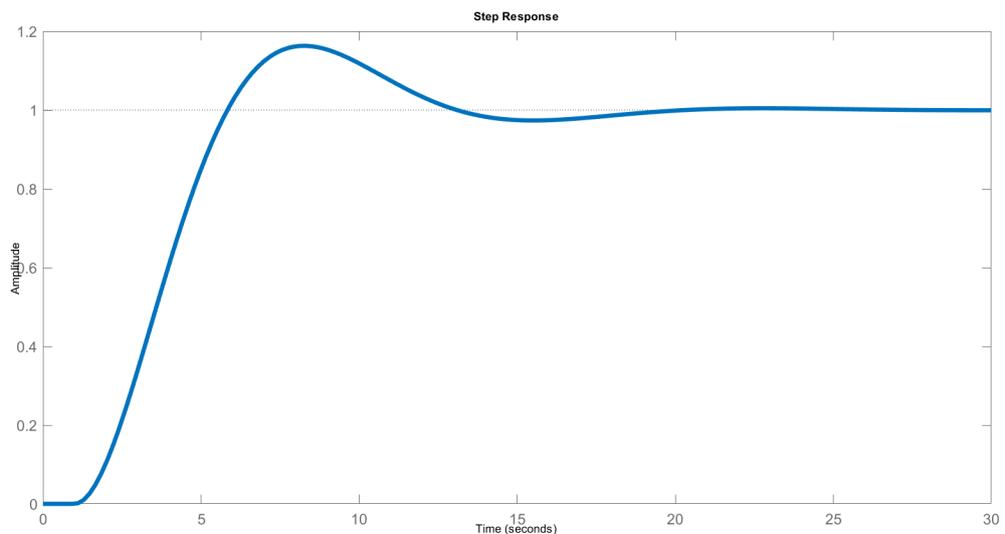
- La période T : la durée d'une oscillation complète du système.
- L'amplitude A : la distance maximale que le système atteint par rapport à son point de consigne.
- Le temps de montée t_r : le temps nécessaire au système pour atteindre 90% de son état d'équilibre après une perturbation.
- Le temps d'installation t_s : le temps nécessaire pour que les oscillations s'éteignent après une perturbation.
- Le facteur d'amortissement ζ : la mesure de l'amortissement du système, c'est-à-dire sa capacité à réduire les oscillations au fil du temps.
- La fréquence naturelle ω_n : la fréquence de résonance du système, qui dépend de sa rigidité et de sa masse.

Le modèle de système du second ordre avec retard, autorégulant ou intégrateur peut être utilisé pour ajuster les réglages du régulateur afin de réduire les oscillations et d'optimiser la réponse du système. Des techniques d'analyse de la réponse fréquentielle peuvent être utilisées pour déterminer les paramètres optimaux pour le régulateur.

Voici un exemple de code MATLAB pour tracer la réponse en fréquence d'un système du second ordre avec retard :

```

% Define the transfer function K = 1; % adjust K as desired T = 1; % adjust T as desired
zeta = 0.5; % adjust zeta as desired wn = 2; % adjust wn as desired num = K * exp(-T *
s); den = 1 + 2*zeta*wn*s + wn^2*s^2; H = num/den;
% Plot the step response step(H)
    
```



3.5 Identification en chaîne fermée

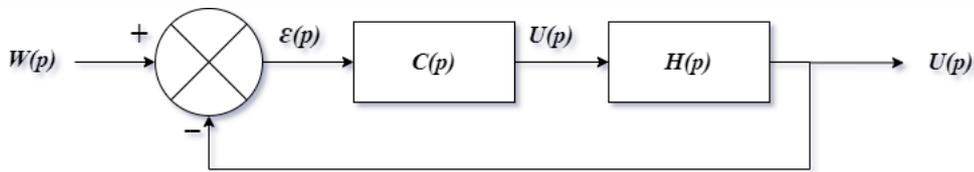
3.5.1 La méthode des oscillations limites

Il y a plusieurs méthodes des oscillations limites, mais en général, cette méthode est utilisée pour estimer les paramètres d'un système de contrôle, tels que les coefficients de l'équation de transfert, à partir de la réponse transitoire du système. Cette méthode utilise des oscillations de plus en plus grandes pour déterminer les limites supérieures et inférieures de l'amplitude de l'oscillation, et ces limites peuvent être utilisées pour estimer les paramètres du système.

Pour appliquer cette méthode, on peut procéder comme suit :

1. Commencer par régler le régulateur de telle sorte que le système soit stable, en utilisant par exemple une méthode de réglage classique (Ziegler-Nichols, Cohen-Coon, etc.).
2. Appliquer une perturbation au système et mesurer la réponse transitoire.
3. Identifier les oscillations dans la réponse et calculer la période de ces oscillations.
4. En utilisant cette période, déterminer les limites supérieure et inférieure de l'amplitude de l'oscillation. Ces limites peuvent être déterminées en utilisant des formules empiriques ou en utilisant des graphes, qui dépendent des caractéristiques de l'oscillation, telles que la période et le rapport de la largeur du pic à la période.
5. En utilisant ces limites, estimer les coefficients de l'équation de transfert du système en utilisant des formules empiriques ou des graphes, qui dépendent des caractéristiques de l'oscillation.
6. Si les estimations des coefficients ne sont pas satisfaisantes, ajuster les paramètres du régulateur et répéter les étapes 2 à 5 jusqu'à ce que les estimations soient satisfaisantes.

Cette méthode peut être répétée pour différents essais de perturbation pour améliorer l'estimation des paramètres du système.



3.5.2 Premier essai

Dans ce cas, le premier essai consisterait à suivre les étapes suivantes :

1. Configurer le régulateur pour être seulement proportionnel, c'est-à-dire $C(p) = K_p$, où K_p est un gain faible, comme $K_p = 1$ ou $0,5$.
2. Si la mesure ne rejoint pas la consigne, il reste un écart statique ou de position . Dans ce cas le procédé est naturellement stable de gain statique ε_p et on montre que : $\varepsilon_p = \frac{A}{1+K K_p}$

3. Si la mesure rejoint la consigne, le procédé étudié est naturellement instable; pour savoir s'il possède une ou deux intégrations on peut produire une rampe de consigne $w = b.t$: (il y a une intégration si l'écart de vitesse en régime permanent est non nul; le gain dynamique K' est calculé à l'aide de la relation : $\varepsilon_v = \frac{b}{KK_p}$ & il y a deux intégrations si l'écart de vitesse en régime permanent est nul, et le gain dynamique K' ne peut être déterminé).
4. Appliquer un échelon de consigne A au système en boucle fermée.
5. Observer la réponse du système et déterminer s'il est naturellement stable ou intégrateur en fonction de sa réponse.

Si la réponse du système est stable et atteint rapidement la valeur nominale sans oscillations significatives, alors le système est considéré comme naturellement stable. Si la réponse oscille ou continue de changer de manière instable, alors le système peut être considéré comme intégrateur.

Il est important de noter que l'identification précise de la réponse du système peut nécessiter plusieurs essais avec différentes configurations de commande et des échelons de consigne de différentes amplitudes. Les résultats de ces essais peuvent être utilisés pour ajuster les paramètres du régulateur afin d'optimiser la réponse du système en boucle fermée.

3.5.3 Deuxième essai

La première étape pour déterminer le gain critique du correcteur K_{lim} et la période d'oscillation T_{OSC} consiste à tracer la réponse en fréquence du système en utilisant la fonction de transfert en boucle ouverte $H(p) = G(p)C(p)$, où $G(p)$ est la fonction de transfert du procédé.

Ensuite, nous devons trouver la fréquence de résonance ω_{OSC} du système, qui correspond à la fréquence où la phase de la fonction de transfert en boucle ouverte atteint -180 degrés et l'amplitude est maximale.

Une fois que la fréquence de résonance est trouvée, nous pouvons déterminer le gain critique K_{lim} en utilisant la condition d'amplitude :

$$|K_{lim}H(j\omega_{OSC})| = 1$$

Le gain K_p doit alors être augmenté progressivement jusqu'à ce que le système commence à osciller. À ce stade, nous pouvons ajuster le gain pour atteindre l'oscillation juste entretenue en utilisant la condition de phase :

$$\arg[K_{lim}H(j\omega_{OSC})] = -180^\circ$$

Lorsque le système atteint l'oscillation juste entretenue, nous pouvons mesurer la période d'oscillation T_{OSC} à partir du signal de sortie enregistré $x(t)$ en utilisant la formule :

$$T_{OSC} = \frac{2\pi}{\omega_{OSC}}$$

où ω_{OSC} est la fréquence de résonance mesurée.

En résumé, la procédure pour déterminer le gain critique du correcteur et la période d'oscillation est la suivante :

Tracer la réponse en fréquence du système en utilisant la fonction de transfert en boucle ouverte $H(p) = G(p)C(p)$. Trouver la fréquence de résonance ω_{OSC} à partir du tracé de la phase de la fonction de transfert en boucle ouverte. Calculer le gain critique K_{lim} en utilisant la condition d'amplitude $|K_{lim}H(j\omega_{OSC})| = 1$. Augmenter progressivement le gain K_p jusqu'à atteindre l'oscillation juste entretenue. Ajuster le gain K_p pour satisfaire la condition de phase $arg[KH(j\omega_{OSC})] = -180^\circ$. Mesurer la période d'oscillation T_{OSC} à partir du signal de sortie enregistré en utilisant la formule $T_{OSC} = \frac{2\pi}{\omega_{OSC}}$.

3.6 Conclusion

En conclusion, l'identification des systèmes en boucle ouverte et fermée est une étape importante dans la conception et la mise en œuvre des systèmes de régulation. Dans les systèmes en boucle fermée, il est important de comprendre la relation entre les entrées et les sorties pour pouvoir concevoir un régulateur qui maintient les sorties à des niveaux souhaités malgré les perturbations. Dans les systèmes en boucle ouverte, il est important de comprendre la relation entre les entrées et les sorties pour pouvoir concevoir un système qui produit les sorties souhaitées.

L'identification des systèmes peut être effectuée à l'aide de diverses techniques telles que l'analyse fréquentielle, l'analyse de réponse impulsionnelle et l'analyse de réponse en échelon. Les résultats de ces analyses peuvent être utilisés pour concevoir des régulateurs qui maintiennent les sorties des systèmes en boucle fermée à des niveaux souhaités et des systèmes en boucle ouverte qui produisent les sorties souhaitées.

Les régulateurs standards : P, PI, PD et PID

4.1 Introduction

Les régulateurs standards P, PI, PD et PID sont des algorithmes de contrôle couramment utilisés pour réguler un système en boucle fermée. Ces régulateurs sont utilisés pour ajuster la sortie d'un système en fonction de la différence entre la valeur de consigne souhaitée et la valeur mesurée. Chacun de ces régulateurs utilise une approche différente pour ajuster la sortie du système et ils sont utilisés dans de nombreuses applications industrielles, notamment dans le contrôle de la température, de la vitesse, de la pression, du débit et d'autres paramètres critiques.

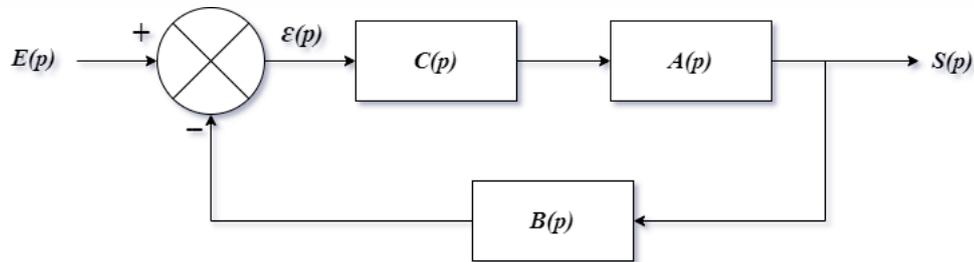
Le choix du régulateur approprié dépend des exigences spécifiques du système et de la précision nécessaire pour atteindre les performances souhaitées. Le régulateur P est le plus simple des régulateurs, tandis que les régulateurs PI et PD sont des variantes plus avancées qui offrent un meilleur contrôle du système. Le régulateur PID est le plus couramment utilisé car il combine les avantages des trois types de régulateurs pour offrir un contrôle précis et stable pour un large éventail de systèmes.

Il est important de comprendre les avantages et les inconvénients de chaque type de régulateur pour déterminer le meilleur choix pour une application spécifique. Les régulateurs standards sont largement utilisés dans l'industrie pour maintenir des processus critiques sous contrôle et pour garantir une production fiable et efficace.

4.2 Principe général de la correction d'un système

Le concept est d'ajouter un correcteur, également appelé dispositif de fonction de transfert supplémentaire $C(p)$, en amont du système $A(p)$, dans la chaîne directe. La tâche

principale d'un correcteur doit être de modifier les performances du système initial.



$$G_c(p) = G_i(p)C(p)$$

où $C(p)$ est la fonction de transfert du correcteur.

$$H_c(p) = \frac{G_c(p)}{1 + G_c(p)H(p)}$$

où $H_i(p)$ est la fonction de transfert en boucle fermée du système initial.

Cela signifie que le correcteur est conçu pour modifier la fonction de transfert en boucle ouverte du système initial $G_i(p)$ en multipliant par $C(p)$, pour obtenir $G_c(p)$.

Ensuite, la fonction de transfert en boucle fermée $H_c(p)$ est calculée en utilisant la fonction $G_c(p)$ et la fonction de transfert en boucle fermée du système initial $H_i(p)$.

4.3 Actions correctives élémentaires

Il existe trois mesures correctives simples qui permettent chacune d'améliorer une certaine performance. Bien qu'ils soient relativement faciles à faire, ils dégradent généralement les autres performances. Lorsque les spécifications ne sont pas trop rigoureuses, elles peuvent être utilisées. Sinon, pensez à fusionner ces plusieurs corrections dans un correcteur plus compliqué.

4.3.1 Correcteur proportionnel

Le correcteur proportionnel est un type de correcteur qui consiste en un simple amplificateur de gain réglable $C(p) = K$. Sa mission est de modifier le gain statique initial du système.

Si $K < 1$, c'est-à-dire s'il s'agit d'un atténuateur, la stabilité du système est améliorée et le dépassement en boucle fermée est réduit. Cependant, la rapidité et la précision du système sont dégradées.

Si $K > 1$, le gain statique initial du système est augmenté, ce qui peut améliorer la rapidité et la précision du système en boucle fermée. Cependant, cela peut également réduire la stabilité du système, voire le rendre instable, et augmenter le dépassement.

Il est important de noter que l'impact d'une variation du gain statique sur les performances en boucle fermée peut être visualisé en se référant au diagramme de Bode d'un système en boucle ouverte.

4.3.2 Correcteur intégral

4.3.2.1 Définition

Le correcteur intégral est un type de correcteur qui ajoute un terme intégrateur à la fonction de transfert du système en boucle fermée. Sa fonction est de corriger l'erreur de régulation à long terme du système.

Le correcteur intégral est souvent utilisé pour éliminer l'erreur de régulation en régime permanent due à des perturbations constantes ou à des écarts constants entre la consigne et la sortie du système.

Mathématiquement, le correcteur intégral est représenté par une fonction de transfert de la forme :

$$C(p) = \frac{K_p}{p}$$

où K_p est le gain intégral.

4.3.2.2 Influence sur les autres performances

Les autres performances du système sont évidemment modifiées par les altérations préférées à la fonction de transfert. C'est encore à nous de déterminer leur évolution. On peut se renseigner sur ces changements en simulant, sur un diagramme de Bode, les transformations que subissent les courbes de gain et de phase une fois ce correcteur ajouté. Considérons n'importe quel système de fonction de transfert en boucle ouverte $G_i(p)$.

Les graphiques montrent respectivement :

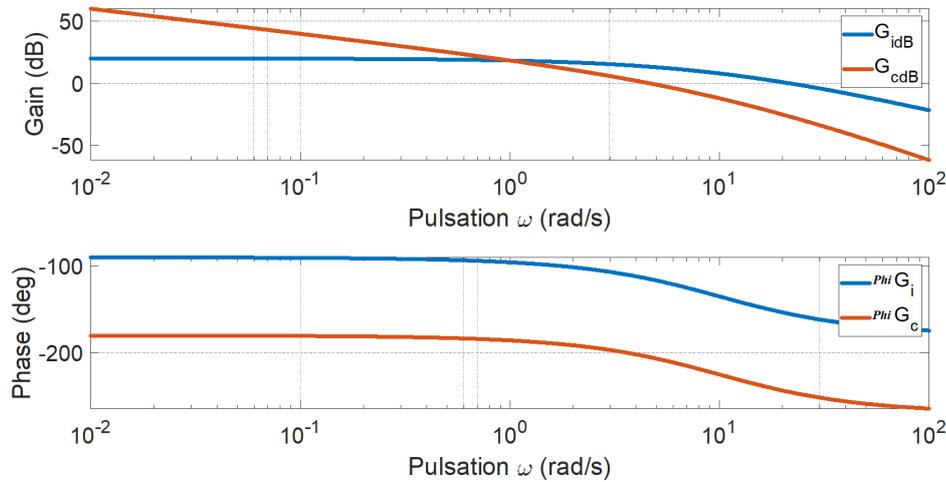
$$\begin{cases} G_{idB} = 20 \log G_i(\omega) \\ \varphi_i(\omega) = \arg G_i(j\omega) \end{cases}$$

Effectivement, les graphes correspondant à la fonction de transfert corrigée se déduisent facilement des graphes initiaux. Pour cela, il suffit de sommer la fonction de transfert corrigée avec la fonction de transfert du correcteur, et de recalculer le gain et la phase à chaque fréquence. Ensuite, il est possible de tracer les courbes de gain et de phase sur un diagramme de Bode pour observer les modifications apportées aux performances du système.

$$\begin{cases} G_{cdB} = 20\log G_c(\omega) = 20\log \frac{G_i(\omega)}{\omega} = 20\log G_i(\omega) - 20\log \omega \\ \varphi_c(\omega) = \arg G_c(j\omega) = \arg \frac{G_i(\omega)}{\omega} = \varphi_i(\omega) - \frac{\pi}{2} \end{cases}$$

Cela signifie que la courbe de gain corrigée G_{cdB} est obtenue en diminuant d'une unité la pente de chaque segment de la courbe de gain initiale G_{idB} , et que le diagramme de phase corrigé est obtenu en décalant la courbe de phase initiale de $\pi/2$ vers le bas. De plus, à la pulsation $\omega = 10$, le gain a chuté de $20dB$, ce qui peut aider à tracer la courbe corrigée.

```
% Courbe de gain initiale w = logspace(-2, 2, 1000); Gi = 10./(1+0.1*w).^2; GidB = 20*log10(abs(Gi));
% Courbe de gain corrigée Gc = Gi./w; GcdB = GidB - 20*log10(w);
% Tracé des courbes de gain subplot(2,1,1) semilogx(w, GidB, 'LineWidth', 1.5) hold on semilogx(w, GcdB, 'LineWidth', 1.5) grid on xlabel('Pulsation \omega (rad/s)') ylabel('Gain (dB)') legend('G_{idB}', 'G_{cdB}')
% Diagramme de phase corrigé subplot(2,1,2) wi = -atan(0.1*w) - pi/2; wc = wi - pi/2; semilogx(w, wi*180/pi, 'LineWidth', 1.5) hold on semilogx(w, wc*180/pi, 'LineWidth', 1.5) grid on xlabel('Pulsation \omega (rad/s)') ylabel('Phase (deg)') legend('\angle G_i', '\angle G_c')
```



On remarque que la pulsation de coupure à 0 dB diminue.

Compte tenu que :

$$t_m \approx \frac{3}{\omega_{c0}}$$

Le temps de montée s'allonge, comme on peut en déduire. Par conséquent, l'intégrateur entraînera généralement le ralentissement du système en boucle fermée.

De plus, étant donné que la courbe de phase s'est décalée et qu'elle finira très probablement très proche de $-\pi$, la marge de phase aura tendance à diminuer même si ω_{c0} a diminué. Détérioration de la stabilité et restriction de dépassement.

En conclusion, l'ajout d'un correcteur d'action inhérent ne fait qu'améliorer la précision du système. Les performances dans tous les autres domaines sont réduites.

4.3.3 Correcteur à action dérivée

Le correcteur dérivateur peut être considéré comme l'inverse du correcteur intégrateur en termes de son effet sur la réponse en fréquence du système. Alors que l'intégrateur ajoute un pôle à l'origine, le dérivateur ajoute un zéro à l'origine. Cela peut être illustré sur un diagramme de Bode.

Supposons que la fonction de transfert en boucle ouverte du système soit $G(p)$. L'ajout d'un correcteur dérivateur $C(p) = p$ donne une fonction de transfert en boucle ouverte de $H(p) = G(p)C(p) = G(p)p$.

Sur un diagramme de Bode, l'ajout d'un correcteur dérivateur ajoute un zéro à l'origine, ce qui a pour effet d'augmenter la pente de la courbe de gain à partir de la fré-

quence de coupure. Si la fonction de transfert en boucle ouverte initiale avait une pente de $-20dB/décade$ avant la fréquence de coupure, l'ajout du correcteur dérivateur donnerait une pente de $-40dB/décade$ après la fréquence de coupure. Cela peut conduire à une augmentation de la bande passante du système et à une amélioration de la réponse transitoire.

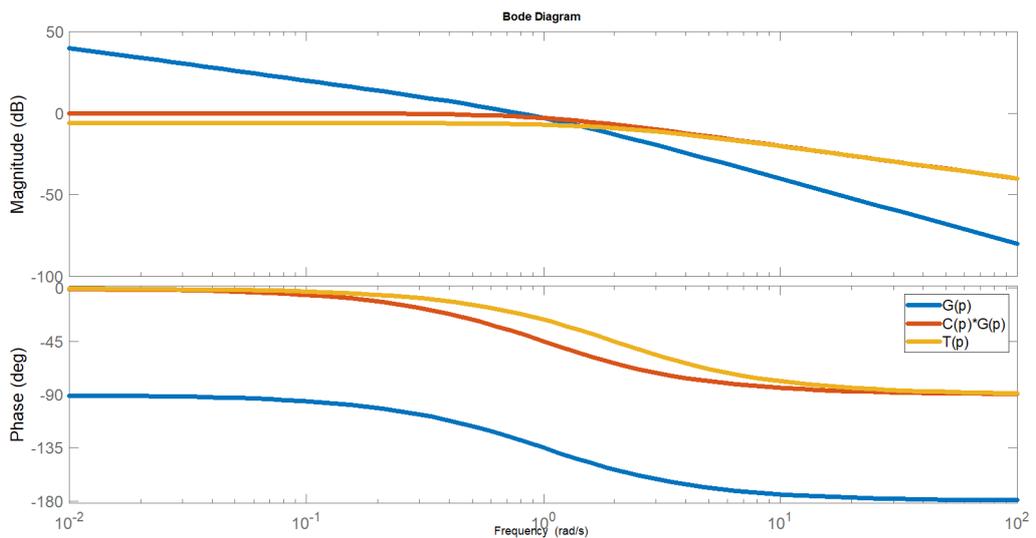
Cependant, il convient de noter que l'ajout d'un correcteur dérivateur peut également amplifier le bruit haute fréquence, ce qui peut rendre le système plus sensible aux perturbations. De plus, l'ajout d'un dérivateur peut rendre le système plus sensible aux erreurs de modélisation, car il amplifie les hautes fréquences. Il est donc important de concevoir soigneusement le correcteur dérivateur pour éviter ces problèmes potentiels.

Voici un exemple de code MATLAB pour tracer l'influence d'un dérivateur sur les performances d'un système de régulation. Ce code utilise la fonction de transfert du système sous la forme d'une expression symbolique, ce qui permet de facilement modifier les paramètres du système et les caractéristiques du correcteur.

```

% Définition de la fonction de transfert du système s = tf('s'); G = 1/(s*(s+1));
% Définition du correcteur dérivateur C = s;
% Définition de la fonction de transfert en boucle fermée T = feedback(C*G,1);
% Tracé des courbes de réponse en fréquence bode(G); hold on; bode(C*G); hold on;
bode(T); legend('G(p)', 'C(p)*G(p)', 'T(p)');

```



Considérons à nouveau un système quelconque de fonction de transfert en boucle ouverte $G_i(p)$. Les graphes représentent respectivement :

$$\begin{cases} G_{idB} = 20 \log G_i(\omega) \\ \varphi_i(\omega) = \arg G_i(j\omega) \end{cases}$$

Les graphes correspondant à la fonction de transfert corrigée se déduisent facilement des graphes initiaux :

$$\begin{cases} G_{cdB} = 20 \log G_c(\omega) = 20 \log \omega G_i(\omega) = 20 \log G_i(\omega) + 20 \log \omega \\ \varphi_c(\omega) = \arg G_c(j\omega) = \arg \omega G_i(\omega) = \varphi_i(\omega) + \frac{\pi}{2} \end{cases}$$

On « ajoute » par conséquent à chaque segment l'équivalent d'un segment de pente [1], c'est-à-dire on augmente chaque pente de début d'une unité, pour passer de la courbe de gain initiale G_{idB} à la courbe corrigée G_{cdB} . Aussi, on peut créer rapidement le graphe correspondant à G_{cdB} en observant qu'à la pulsation $\omega = 10$, le gain a augmenté de $20dB$. Pendant ce temps, le diagramme de phase est translaté vers le haut de $\pi/2$.

On remarque que la pulsation de coupure à $0dB$ augmente.

Compte tenu que :

$$t_m \approx \frac{3}{\omega_{c0}}$$

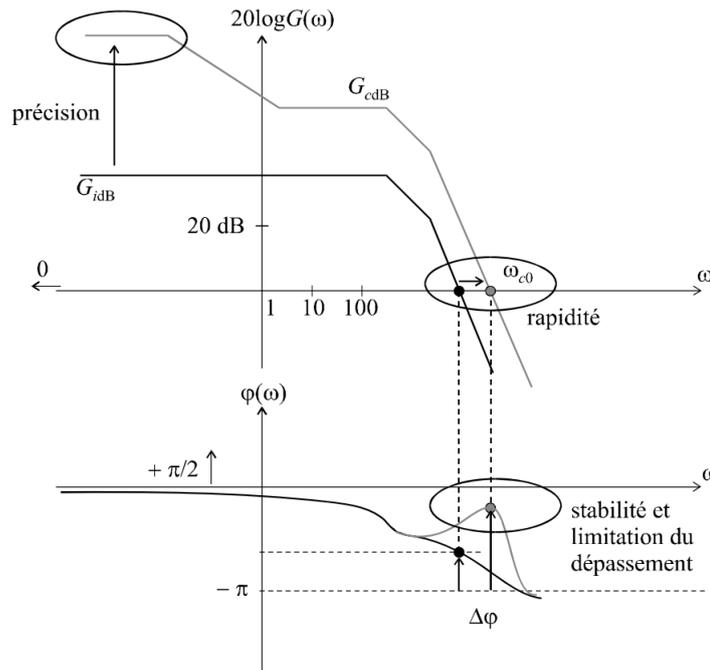
On peut en déduire que le temps de montée diminue. Le dérivateur aura donc tendance à accélérer le système en boucle fermée.

La marge de phase est également affectée par une augmentation de ω_{c0} , mais cette influence est dépendante de l'ordre du système. La « montée en phase de $+\pi/2$ » peut en effet avoir deux issues différentes : si le système est d'ordre élevé, le déphasage peut tendre vers des valeurs négatives extrêmement importantes ; dans ce cas, l'élévation de phase peut n'avoir aucun effet sur l'amélioration de la marge de phase voire la dégrader et rendre le système instable. Si l'ordre du système est bas, par contre, l'augmentation de phase peut produire une nouvelle courbe, $\varphi_c(\omega)$, qui tend vers une valeur considérablement supérieure à $-\pi$.

Enfin, comme le gain aux basses fréquences diminue drastiquement, l'action dérivée va nuire à la précision du système, qui est liée au gain statique. En conclusion, l'ajout d'un correcteur d'action dérivée ne fait qu'augmenter la rapidité du système. Toutes les autres performances sont soit compromises, soit susceptibles de l'être.

4.4 Inconvénient fondamental des actions correctives élémentaires

Reprenons le diagramme de Bode étudié précédemment et essayons d’imaginer quel serait l’action d’un correcteur idéal.



Le gain statique ne devrait pouvoir être augmenté que dans le domaine des basses fréquences afin d’améliorer la précision. La situation idéale serait de ne pouvoir modifier la courbe de phase autour de ω_{c0} que pour augmenter la marge de phase et améliorer la vitesse ; toutes ces modifications devraient bien sûr être localisées à seulement 8,5 actions proportionnelles intégrales. Correcteur de retard de phase 153° dans certaines gammes de fréquences. Parce que chacune des mesures correctives de base dont nous venons de parler a un gros défaut : leur action s’étend sur toute la plage de fréquences de 0 à l’infini, et toute mesure corrective visant à améliorer une performance à un moment donné affecte également d’autres emplacements en abaissant les performances de certains. Pour corriger l’une des performances sans affecter les autres, les correcteurs idéaux, s’il en existe, devront présenter une action limitée. C’est quelque chose que les correcteurs de base qui modifient l’ensemble du diagramme de Bode ne peuvent pas faire.

4.5 Action proportionnelle intégrale correcteur à retard de phase

4.5.1 Définition

Le correcteur à retard de phase est un correcteur utilisé en régulation et en automatique pour améliorer la précision d'un système asservi en augmentant le gain aux basses fréquences. Il est appelé "à retard de phase" car il ajoute un délai de phase à la réponse en fréquence du système.

Sa fonction de transfert est :

$$C(p) = \frac{a(1 + Tp)}{1 + aTp}$$

avec $a > 1$

Pour mieux comprendre l'action de ce correcteur, traçons son diagramme de Bode. Il y a deux pulsations de coupure : $1/T$ et $1/aT$

telles que : $1/aT < 1/T$

On a :

$$C(\omega) = \frac{a\sqrt{1 + T^2\omega^2}}{\sqrt{1 + a^2T^2\omega^2}}$$

et :

$$\varphi(\omega) = \arctan T\omega - \arctan aT\omega$$

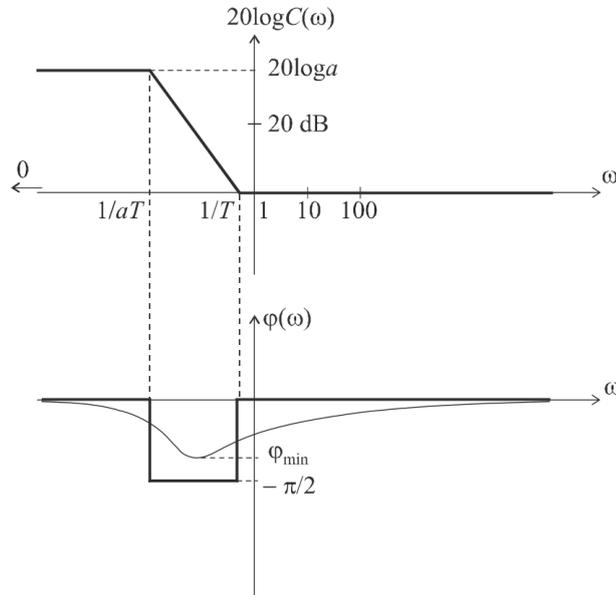
Lorsque $\omega \rightarrow 0$, on a : $C(\omega) \rightarrow a$

Cet équivalent de pente nulle est valable de 0 jusqu'à la première pulsation de coupure qui a pour expression : $1/aT$. La pente du diagramme de Bode asymptotique se décrémente alors d'une unité et ce nouvel équivalent est valable jusqu'à la seconde pulsation de coupure ($1/T$) à partir de laquelle nous retrouvons une pente nulle.

Lorsque $\omega \rightarrow +\infty$, on a : $20\log C(\omega) \rightarrow 0dB$

L'examen du diagramme de Bode nous permet de prévoir l'action de ce correcteur. Lorsque celui-ci sera placé en cascade avec le système à corriger, dans la chaîne directe, les deux diagrammes de Bode s'additionneront. Le gain statique est donc bien augmenté de $20\log a$, ce qui améliore la précision. En réglant le paramètre T sur une valeur suffisamment faible, cette correction n'a d'influence qu'aux basses fréquences ; le gain aux hautes fréquences n'est pratiquement pas affecté. Le déphasage négatif supplémentaire introduit par le correcteur se situe également aux basses fréquences. Il n'a donc pas d'influence sur la marge de stabilité, étant donné que les pulsations de coupure à $0dB$ sont, en général, situées dans des plages de fréquences plus élevées. En tout état de cause, pour régler le correcteur

à retard de phase, on choisira la valeur de a qui permet d'obtenir le gain statique résultant voulu et on choisira ensuite T se sorte que $1/T \ll \omega_{c0}$.



4.5.2 Exemple

Considérons un système de fonction de transfert $G(p)$ placé dans une boucle à retour unitaire, avec :

$$G(p) = \frac{K}{\left(1 + \frac{p}{10}\right)^3}$$

Le paramètre K , gain statique du système en boucle ouverte est positif et réglable. On souhaite que ce système présente en boucle fermée une erreur de position $\varepsilon_p = 5\%$, tout en ayant une marge de phase $\Delta\varphi = 45^\circ$.

On commence par régler K pour satisfaire à la condition sur la marge de phase :

Comme :

$$G(p) = \frac{K}{\left(1 + \frac{j\omega}{10}\right)^3}$$

on a :

$$\Delta\varphi = \pi - 3\arctan\frac{\omega_{c0}}{10} = \frac{\pi}{4}$$

soit : $\omega_{c0} = 10\text{rad/s}$

On a donc :

$$G(\omega c0) = \frac{K}{\left(\sqrt{1 + \frac{\omega_{c0}^2}{100}}\right)^3} = 1$$

d'où : $K = (\sqrt{2})^3 = 2,8 \Rightarrow 20\log K = 8,9dB$

Calculons à présent l'erreur de position obtenue en boucle fermée dans ces conditions :

$$\varepsilon_p = \lim_{p \rightarrow 0} [1 - H(p)] = \lim_{p \rightarrow 0} \left[1 - \frac{K}{K + \left(1 + \frac{p}{10}\right)^3} \right]$$

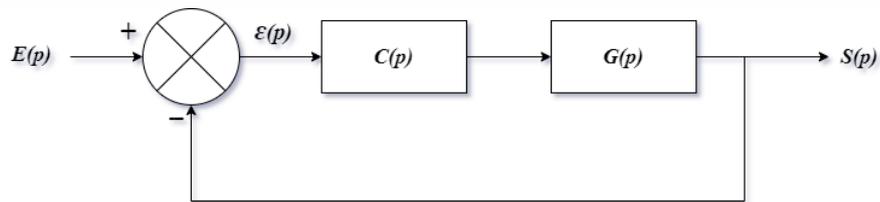
soit :

$$\varepsilon_p = \frac{1}{1 + K} = 0,26 = 26$$

La précision constatée ne satisfait pas au cahier des charges. Pour obtenir une erreur de position de 5 %, il est nécessaire de disposer d'un gain statique K' tel que :

$$\varepsilon_p = \frac{1}{1 + K'} = 0,05 = 5 \Rightarrow K' = 19 \Rightarrow 20\log K' = 25,6dB$$

Introduisons un correcteur à retard de phase dans la chaîne directe.



On a :

$$C(p) = \frac{a(1 + Tp)}{1 + aTp}$$

avec $a > 1$

Le nouveau gain statique est : $K' = 2,8a$.

Par conséquent, il est nécessaire de régler le paramètre a de sorte que :

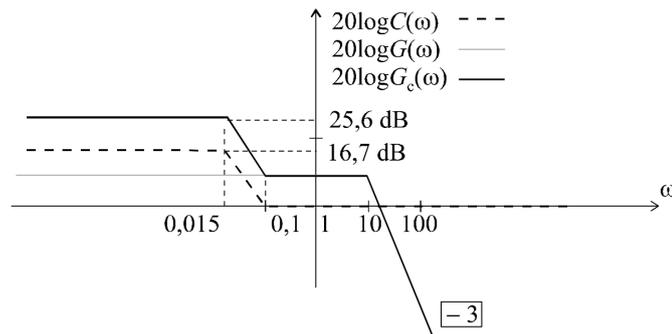
$$a = \frac{19}{2,8} = 6,8 \Rightarrow 20\log a = 16,7dB$$

Il suffit, pour finir, de choisir T de manière à ce que $1/T$ soit très inférieur à la pulsation de coupure à $0dB$. Nous pouvons prendre, par exemple, $T = 10s$.

On a finalement :

$$C(p) = \frac{6,8(1 + 10p)}{1 + 68p}$$

La présente les diagrammes de Bode comparés du système initial et du système corrigé. Rappelons que les diagrammes de Bode de $G(p)$ et de $C(p)$ s'additionnent pour former celui du système corrigé $G_c(p)$.



4.6 Action proportionnelle dérivée correcteur à avance de phase

4.6.1 Définition

Le correcteur à avance de phase est utilisé pour augmenter la marge de phase d'un système en introduisant une avance de phase dans la boucle de rétroaction. L'objectif est de compenser un déphasage insuffisant autour de la pulsation de coupure à $0dB$, ce qui peut entraîner une instabilité ou des oscillations dans le système. Le correcteur à avance de phase est généralement utilisé pour améliorer les performances de stabilité et de temps de réponse des systèmes de contrôle.

On prend :

$$C(p) = \frac{1 + aTp}{1 + Tp}$$

avec $a > 1$

Pour mieux comprendre l'action de ce correcteur, traçons son diagramme de Bode. Il y a deux pulsations de coupure : $1/T$ et $1/aT$, avec :

$$1/T > 1/aT$$

On a :

$$C(\omega) = \frac{\sqrt{1 + a^2 T^2 \omega^2}}{\sqrt{1 + T^2 \omega^2}}$$

$$\text{et } \varphi(\omega) = \arctan(aT\omega) - \arctan T\omega$$

Lorsque $\omega \rightarrow 0$, on a : $C(\omega) \rightarrow 1$

Cet équivalent de pente nulle est valable de 0 jusqu'à la première pulsation de coupure qui a pour expression : $1/aT$. La pente du diagramme de Bode asymptotique s'incrémente alors d'une unité et ce nouvel équivalent est valable jusqu'à la seconde pulsation de coupure ($1/T$) à partir de laquelle nous retrouvons une pente nulle.

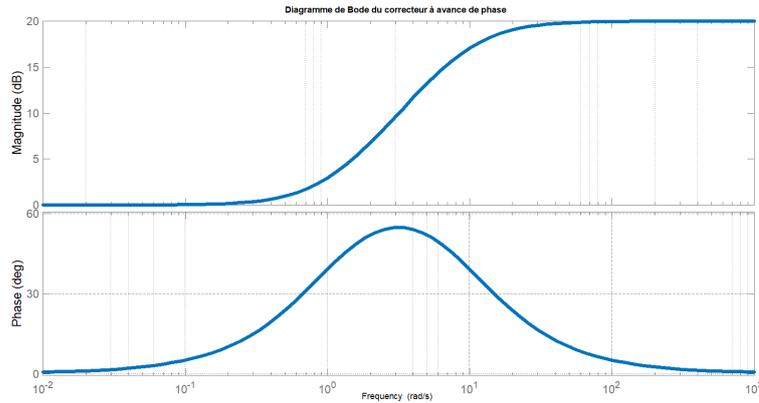
Lorsque $\omega \rightarrow +\infty$, on a : $20 \log C(\omega) \rightarrow 20 \log a$

L'intérêt de ce correcteur est visible sur son diagramme de phase : à la pulsation $\omega_{max} = \frac{1}{T\sqrt{a}}$, le déphasage présente un maximum que nous pouvons facilement calculer :

$$\varphi_{max} = \arcsin \frac{a - 1}{a + 1}$$

Le code MATLAB pour tracer le diagramme de Bode du correcteur est le suivant :

```
>> % Définition du correcteur à avance de phase Td = 0.1; % constante de temps du filtre
    dérivé Ta = 1; % constante de temps du filtre intégral Gc = tf([Ta 1], [Td 1]);
% Calcul du diagramme de Bode du correcteur bode(Gc); grid on; title('Diagramme de
Bode du correcteur à avance de phase');
```



Le principe de l'action corrective consiste à faire coïncider v_{max} avec la pulsation de coupure à $0dB$ ω_{c0} du système à corriger et à régler w_{max} , que l'on appelle la remontée de phase, de manière à obtenir la marge de phase voulue.

4.6.2 Exemple

Considérons un système de fonction de transfert $G(p)$ placé dans une boucle à retour unitaire, avec :

$$G(p) = \frac{100}{(p + 1)^2}$$

On souhaite corriger ce système de manière à ce que sa marge de phase soit égale à 45° . Calculons sa marge de phase avant correction.

On a :

$$G(\omega) = \frac{100}{1 + \omega_{c0}^2} = 1$$

d'où : $\omega_{c0} = \sqrt{99} = 9,95rad/s$

Par conséquent : $\Delta\varphi = \pi - 2arctan\omega_{c0} = 0,2rad = 11^\circ$

La marge de phase est insuffisante. Pour la corriger, nous devons procéder à une remontée de phase de 34° à la pulsation ω_{c0} . On introduit donc un correcteur à avance de phase que l'on règle de manière à ce que :

$$\begin{cases} \frac{1}{T\sqrt{a}} = \omega_{c0} = 9,95rad/s \\ \varphi_{max} = 34^\circ = arcsin\frac{a-1}{a+1} \end{cases}$$

On a donc :

$$\arcsin \frac{a-1}{a+1} = 34^\circ \Rightarrow a = \frac{1 + \sin 34^\circ}{1 - \sin 34^\circ} = 3,54$$

ou encore :

$$20 \log a = 11 \text{ dB}$$

puis :

$$\frac{1}{T\sqrt{a}} = \omega_{c0} \Rightarrow T = \frac{1}{9,95\sqrt{3,54}} = 0,053 \text{ s}$$

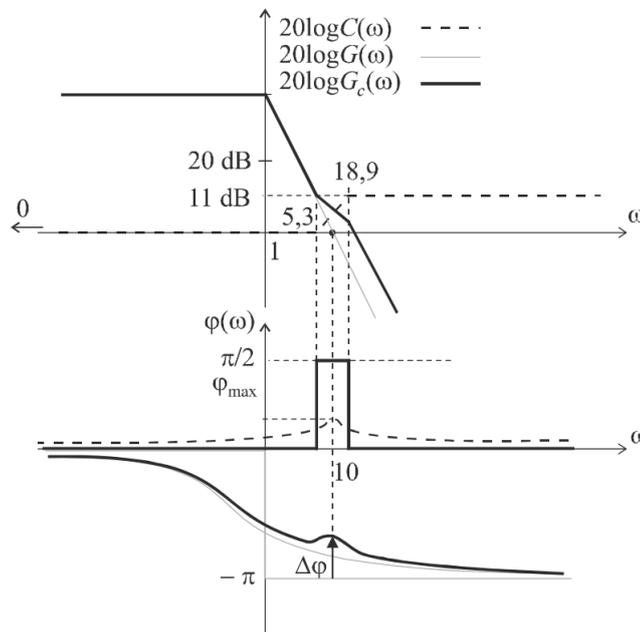
$$\text{soit : } \begin{cases} \frac{1}{T} = 18,9 \text{ rad/s} \\ \frac{1}{aT} = 5,3 \text{ rad/s} \end{cases}$$

Finalement :

$$C(p) = \frac{1 + 0,19p}{1 + 0,053p}$$

La nouvelle fonction de transfert en boucle ouverte est :

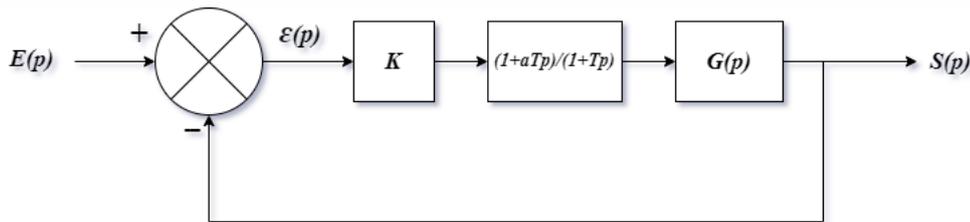
$$G_c(p) = C(p)G(p) = \frac{1 + 0,19p}{1 + 0,053p} \cdot \frac{100}{(p+1)^2}$$



La présente les diagrammes de Bode comparés du système initial et du système corrigé. Rappelons que les diagrammes de Bode de $G(p)$ et de $C(p)$ s'additionnent pour former celui du système corrigé $G_c(p)$. Dans la cas du correcteur à avance de phase, l'action corrective est parfaitement visible sur le diagramme de phase.

4.6.3 Stratégie de réglage d'un asservissement

Les exigences d'une boucle de régulation imposent typiquement 4 performances : la vitesse, matérialisée par une valeur maximale de temps de montée $t_m < seuil$; la précision, matérialisée par une valeur minimale de marge de phase $\Delta\varphi < seuil$; et la marge de stabilité, matérialisée par une valeur maximale d'erreur de position : $\varepsilon_p < seuil$ étant donné que $\zeta_{BF} \approx \frac{\Delta\varphi^\circ}{100}$ et que la limitation de dépassement : $d\% < seuil$ se traduisant par une valeur minimale du coefficient d'amortissement boucle fermée, par une valeur minimale de la marge de phase. Entre cette valeur et celle dictée précédemment, on prend bien sûr la plus élevée.



Une des méthodes les plus intéressantes pour régler le système de sorte qu'il satisfasse au cahier des charges est la suivante : On commence par corriger la précision et la rapidité, soit en réglant le gain statique, soit en ajoutant un correcteur proportionnel. Si la précision parfaite est exigée, on introduit un intégrateur et on règle ensuite la rapidité. Le réglage de la précision et de la rapidité a pour conséquence une détérioration de la marge de phase. On estime alors la valeur de la marge de phase et on introduit un correcteur à avance de phase qui remonte $\Delta\varphi$ à la valeur voulue.

4.7 Correcteur Proportionnel Intégral Dérivé, PID

Le régulateur PID (Proportionnel Intégral Dérivé), qui permet les performances (amortissement, temps de réponse, etc.) d'un procédé modélisé par un second ordre, est le ré-

gulateur standard le plus utilisé dans l'industrie. Plusieurs systèmes physiques compliqués présentent un comportement ressemblant à celui d'un second ordre.

Ainsi, lorsque l'on n'est pas trop exigeant sur les performances de la boucle fermée par rapport à celles de la boucle ouverte, le contrôleur PID est bien adapté à la majorité des procédés de type industriel et est plutôt robuste vis-à-vis des fluctuations des paramètres du procédé (par exemple, très accélération importante de la réponse ou augmentation très importante de l'amortissement en boucle fermée).

En réalité, le type de correcteur choisi concerne une certaine catégorie de systèmes à gérer. Connaître les résultats de diverses actions, y compris les actions proportionnelles, intégrales et dérivées, est crucial pour prendre une décision sensée.

Un régulateur PID est obtenu par l'association de ces trois actions et il remplit essentiellement les trois fonctions suivantes :

1. Il fournit un signal de commande en tenant compte de l'évolution du signal de sortie par rapport à la consigne.
2. Il élimine l'erreur statique grâce au terme intégrateur.
3. Il anticipe les variations de la sortie grâce au terme dérivateur.

4.7.1 Structure des correcteur PID

On distingue trois structures pour les correcteurs PID :

4.7.1.1 Structure série

L'expression temporelle qui relie la commande $u(t)$ avec l'erreur $\varepsilon(t)$ de cette structure est donné par :

$$\begin{cases} u(t) = K_p \varepsilon(t) + T_d \frac{d\varepsilon(t)}{dt} + \frac{1}{T_i} \int_0^t \varepsilon(t) dt \\ C(p) = K_p + T_d p + \frac{1}{T_i p} \end{cases}$$

4.7.1.2 Structure parallèle

La fonction de transfert de correcteur PID série peut être déduite comme suit :

$$C(p) = \frac{U(p)}{\varepsilon(p)} = K_p (1 + T_d p) \left(1 + \frac{1}{T_i p}\right)$$

4.7.1.3 Structure mixte

La fonction de transfert de correcteur PID mixte est donnée par :

$$C(p) = \frac{U(p)}{\varepsilon(p)} = K_p \left(1 + T_d p + \frac{1}{T_i p} \right)$$

4.7.2 Méthodes de synthèse

La synthèse de correcteurs est utilisée pour résoudre le problème de détermination des correcteurs pour les systèmes de contrôle. Il existe de nombreuses méthodes de synthèse disponibles, mais leur classification précise est difficile à réaliser. Toutefois, dans cette section, deux types de méthodes sont distingués : les méthodes de conception basées sur le temps et les méthodes de conception basées sur la fréquence.

Il existe deux types de méthodes pour la détermination des paramètres de régulateur : les méthodes empiriques et les méthodes basées sur la connaissance du modèle du système.

- Les méthodes empiriques ne nécessitent pas une connaissance parfaite du modèle du système à contrôler, mais elles reposent sur les observations expérimentales telles que la réponse indicielle. Ces méthodes sont simples et largement utilisées dans l'industrie, mais elles ne permettent pas un réglage fin du régulateur.
- Les méthodes basées sur la connaissance du modèle du système, par exemple sous forme de fonction de transfert, permettent de calculer les paramètres du régulateur de manière à obtenir la fonction de transfert souhaitée en boucle ouverte ou en boucle fermée. Ces méthodes sont plus précises mais nécessitent une connaissance préalable du modèle du système.

4.7.2.1 Méthodes empiriques

En 1942, Ziegler et Nichols ont introduit deux méthodes heuristiques pour régler rapidement les paramètres des correcteurs P, PI et PID, qui s'appuient sur leur expérience et des simulations limitées. La première méthode implique la mesure de la réponse indicielle en boucle ouverte, tandis que la seconde méthode requiert de pousser le système bouclé à sa limite de stabilité.

- **Méthode de Ziegler-Nichols en boucle ouverte**

La méthode de Ziegler-Nichols en boucle ouverte ne permet pas de calculer directement la fonction de transfert du correcteur $C(p)$, mais plutôt de déterminer les paramètres optimaux du correcteur proportionnel (P), proportionnel-intégral (PI) ou proportionnel-intégral-dérivé (PID) en se basant sur la réponse indicielle du système en boucle ouverte.

Les paramètres optimaux peuvent ensuite être utilisés pour construire la fonction de transfert du correcteur $C(p)$. Les formules pour le calcul des paramètres optimaux varient selon le type de correcteur, mais elles dépendent toutes de la réponse indicielle du système en boucle ouverte. Il n'y a donc pas une équation unique pour la fonction de transfert $C(p)$ dans la méthode de Ziegler-Nichols en boucle ouverte.

$$G(p) = \frac{K e^{-\tau p}}{(T_s p + 1)}$$

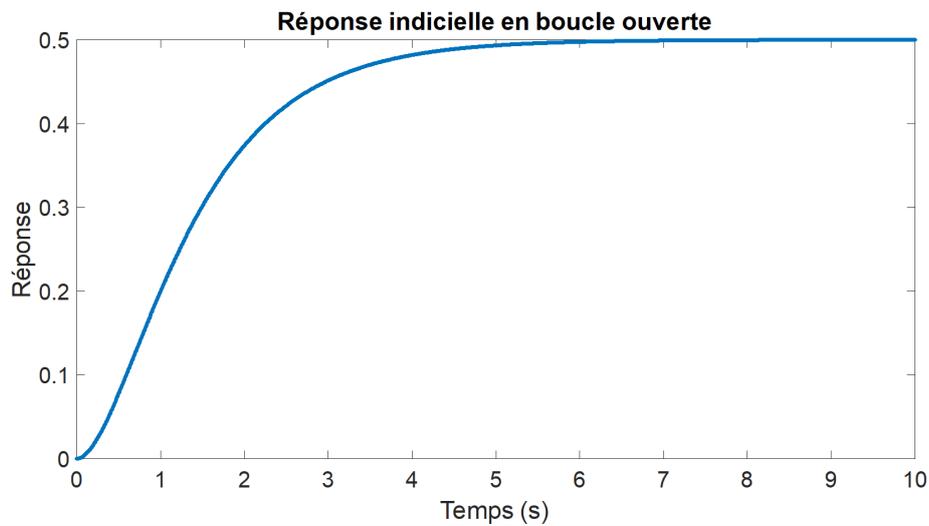
Cette fonction de transfert $G(p)$ d'un système du premier ordre avec un retard pur, où K est le gain statique, T_s est la constante de temps et τ est le retard. La méthode de Ziegler-Nichols en boucle ouverte ne permet pas de calculer directement la fonction de transfert $G(p)$, mais plutôt de déterminer les paramètres optimaux du correcteur proportionnel (P), proportionnel-intégral (PI) ou proportionnel-intégral-dérivé (PID) en se basant sur la réponse indicielle du système en boucle ouverte. Les formules pour le calcul des paramètres optimaux varient selon le type de correcteur, mais elles dépendent toutes de la réponse indicielle du système en boucle ouverte.

Voici un exemple d'implémentation d'un contrôleur PID dans MATLAB à l'aide de la fonction de transfert :

```

% Plant transfer function num = 1; % Numerator coefficients den = [1 3 2]; %
Denominator coefficients G = tf(num, den); % Plant transfer function
% Open-loop step response t = 0:0.01:10; u = ones(size(t)); y = lsim(G, u, t); plot(t, y);
xlabel('Time (s)'); ylabel('Output'); title('Open-loop step response of plant');
% Ziegler-Nichols open-loop tuning method [Kp, Tp] = ziegler_nichols_open_loop(G);
% Proportional controller transfer function C = tf(Kp, 1);
% Closed-loop system transfer function sys_cl = feedback(C*G, 1);
% Step response of the closed-loop system t = 0:0.01:10; u = ones(size(t)); [y, t] =
lsim(sys_cl, u, t); plot(t, y); xlabel('Time (s)'); ylabel('Output'); title('Step response
of PID controlled system');
function [Kp, Tp] = ziegler_nichols_open_loop(G) % Ziegler-Nichols open-loop tuning
method [y, t] = step(G); [ymax, iymax] = max(y); Tp = t(iymax); Kp = 0.5*ymax/Tp;
end

```



• **Méthode de Ziegler-Nichols en boucle fermée**

La méthode de Ziegler-Nichols en boucle fermée consiste à augmenter le gain du correcteur proportionnel jusqu'à obtenir une oscillation soutenue du système. On note alors la valeur critique du gain K_c et la période d'oscillation P_c . On peut ensuite utiliser le tableau suivant pour déterminer les gains K_p , K_i et K_d du régulateur PID :

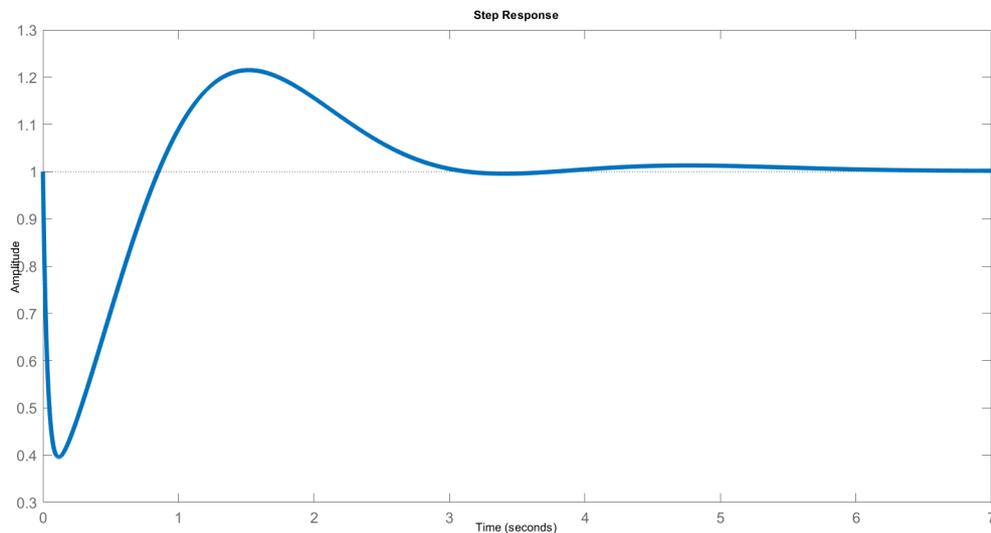
Type de régulateur	K_p	K_i	K_d
P	$0.5 * K_c$	-	-
PI	$0.45 * K_c$	$1.2 * K_p / P_c$	-
PID	$0.6 * K_c$	$2 * K_p / P_c$	$K_p * P_c / 8$

Voici un exemple de code Matlab pour la méthode de Ziegler-Nichols en boucle fermée pour un système de contrôle proportionnel-intégral-dérivé (PID) :

```

% Paramètres du système Kp = 1; Ki = 0.5; Kd = 0.1; tau = 0.5;
% Fonction de transfert du système sys = tf([Kd Kp Ki], [tau 1 0]);
% Méthode de Ziegler-Nichols en boucle fermée T = feedback(sys,1); % Boucle fermée
step(T);
% Calcul des gains du PID selon la méthode de Ziegler-Nichols Kp_zn = 1.2*(tau/Kp);
Ki_zn = 2*Kp/tau; Kd_zn = 0.5*Kp*tau;
% PID avec gains de la méthode de Ziegler-Nichols PID_zn = pid(Kp_zn,Ki_zn,Kd_zn);
T_zn = feedback(PID_zn*sys,1); % Boucle fermée step(T_zn);

```



4.7.2.2 Méthodes algébriques

Les méthodes algébriques, également appelées méthodes théoriques, sont des méthodes de synthèse de commande qui reposent sur la connaissance précise d'un modèle mathématique du système à contrôler. Ces méthodes permettent de déterminer les caractéristiques du correcteur qui permettront de commander efficacement le système.

Les performances réelles obtenues dépendent de la qualité du modèle mathématique utilisé pour représenter le système. Il est donc important d'avoir un modèle précis et fidèle au comportement réel du système pour obtenir des performances satisfaisantes.

Parmi les méthodes algébriques, on trouve notamment la méthode de placement de pôles, qui consiste à placer les pôles de la fonction de transfert en des positions spécifiques pour obtenir les performances souhaitées, et la méthode du retour d'état, qui consiste à utiliser les sorties du système et les mesures de l'état interne pour calculer les signaux de commande.

En somme, les méthodes algébriques sont des méthodes de synthèse de commande qui permettent d'obtenir des performances optimales en utilisant des modèles mathématiques précis du système à contrôler. Ces méthodes sont souvent utilisées en génie électrique, en génie mécanique et dans d'autres domaines de l'ingénierie pour contrôler des systèmes complexes.

• Méthode de placement des pôles

Voici un exemple de calcul utilisant la méthode de placement des pôles pour concevoir un correcteur pour un système de contrôle :

Supposons que nous avons un système de contrôle dont la fonction de transfert en boucle ouverte est donnée par :

$$G(p) = \frac{p + 1}{p^2 + 4p + 5}$$

Nous voulons concevoir un correcteur qui place les pôles de la fonction de transfert en boucle fermée du système à des positions spécifiques pour obtenir un temps de réponse de 2 secondes et un facteur d'amortissement de 0,7.

Tout d'abord, nous calculons les pôles désirés en utilisant les spécifications de conception. Pour un temps de réponse de 2 secondes et un facteur d'amortissement de 0,7, les pôles désirés sont donnés par :

$$\begin{cases} p_1 = -2.85 + j1.99 \\ p_2 = -2.85 - j1.99 \end{cases}$$

Ensuite, nous calculons les coefficients du correcteur à partir des pôles désirés. La fonction de transfert du correcteur peut être écrite sous la forme générale :

$$C(p) = \frac{K(p - z_1)(p - z_2)}{(p - p_1)(p - p_2)}$$

Où K est le gain du correcteur et z_1 et z_2 sont les zéros du correcteur. En utilisant les pôles désirés, nous pouvons déterminer les valeurs de K , z_1 et z_2 de la manière suivante :

$$K = 1 / (G(0) * abs((p_1 - p_2) / (conj(p_1) - p_1))) z_1 = -3z_2 = -2$$

Une fois que nous avons déterminé les coefficients du correcteur, nous pouvons calculer la fonction de transfert en boucle fermée du système en utilisant la formule :

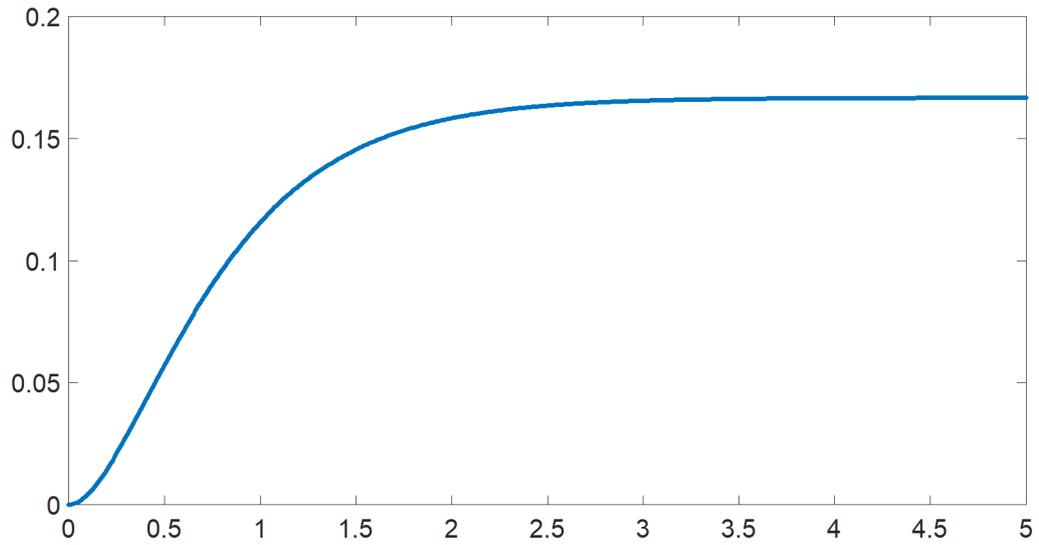
$$T(p) = G(p) * C(p) / (1 + G(p) * C(p))$$

En utilisant la fonction de transfert en boucle fermée, nous pouvons vérifier que les pôles du système sont bien placés aux positions désirées. Enfin, nous pouvons simuler le système pour vérifier que les performances du système sont conformes aux spécifications de conception.

Cet exemple montre comment la méthode de placement des pôles peut être utilisée pour concevoir un correcteur pour un système de contrôle en plaçant les pôles de la fonction de transfert en boucle fermée aux positions désirées pour obtenir les performances souhaitées.

Voici un autre exemple de code MATLAB pour le placement des pôles d'un système en utilisant la fonction 'place' :

```
% Définition des paramètres du système A = [0 1; -4 -5]; B = [0; 1]; C = [1 0]; D = 0;
sys = ss(A, B, C, D); % Système en espace d'état
% Calcul des pôles actuels du système poles = eig(A);
% Placement des pôles desired_poles = [-2; -3]; K = place(A, B, desired_poles); % Calcul
de la matrice de gain
% Affichage des pôles new_poles = eig(A - B*K)
% Simulation du système en boucle fermée sys_cl = ss(A - B*K, B, C, D); % Système en
boucle fermée t = 0:0.01:5; % Temps de simulation u = ones(size(t)); % Entrée unitaire
[y, ~, ~] = lsim(sys_cl, u, t); % Réponse du système plot(t, y); % Affichage de la réponse
```



• **Méthode de compensation des pôles**

a méthode de compensation des pôles est une technique utilisée en électronique pour stabiliser les systèmes à rétroaction. Elle consiste à ajouter des composants qui compensent les effets de résonance ou de distorsion causés par les pôles du système. Voici un exemple de calcul pour la méthode de compensation des pôles :

Supposons que nous avons un système à rétroaction avec une fonction de transfert $H(p)$ donnée par :

$$H(p) = \frac{K}{(p^2 + ap + b)}$$

où K , a et b sont des constantes. Ce système a deux pôles, qui sont donnés par les solutions de l'équation :

$$sp^2 + ap + b = 0$$

Les pôles peuvent être réels ou complexes, selon les valeurs de a et b . Si les pôles sont proches de l'axe imaginaire, le système peut être instable et produire des oscillations indésirables.

Pour compenser les effets de ces pôles, nous pouvons ajouter un composant $C(p)$ à la fonction de transfert $H(p)$, de sorte que la nouvelle fonction de transfert soit donnée par :

$$H(p) = \frac{KC(p)}{(p^2 + ap + b)}$$

Le but est de choisir le composant $C(p)$ de manière à ce que les pôles de la nouvelle fonction de transfert soient déplacés vers des positions plus stables. Une méthode courante pour cela est de choisir $C(p)$ sous la forme :

$$C(p) = \frac{p + c}{p + d}$$

où c et d sont des constantes à choisir de manière appropriée. En substituant cette expression dans la fonction de transfert $H(p)$, on obtient :

$$H(p) = \frac{K(p + c)}{(p + d)(p^2 + ap + b)}$$

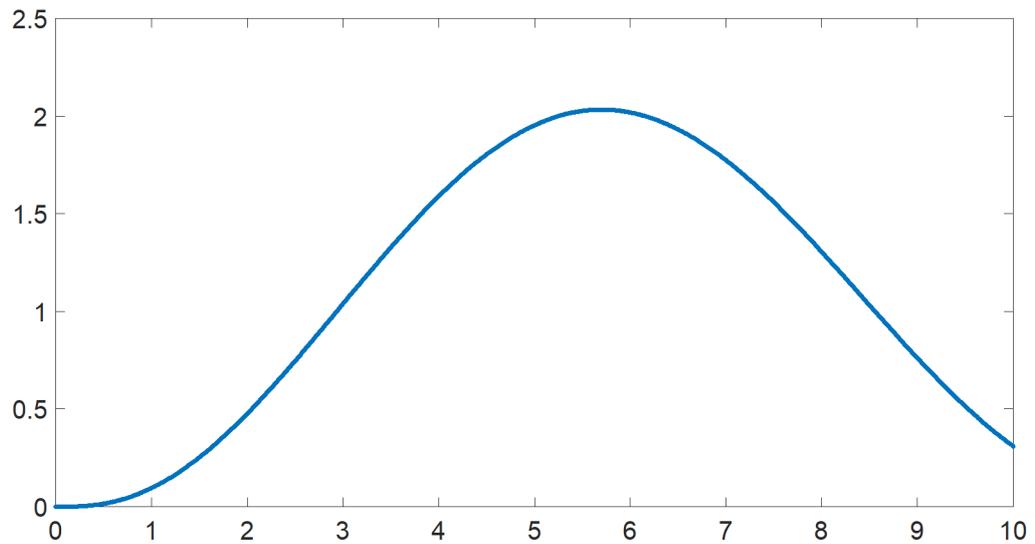
Les pôles de la nouvelle fonction de transfert sont donnés par les solutions de l'équation :

$$(s + d)(s^2 + as + b) = 0$$

En choisissant judicieusement les valeurs de c et d , on peut déplacer les pôles vers des positions plus stables et donc stabiliser le système à rétroaction. Le choix optimal de c et d dépend des caractéristiques du système et doit être effectué par analyse mathématique ou par simulation numérique.

```

% Définition des paramètres du système s = tf('s'); G = 1 / (s*(s+1)*(s+2)); % Fonction
de transfert
% Calcul des pôles actuels du système poles = pole(G);
% Définition du correcteur PI Kp = 1; Ki = 0.5; C = pid(Kp, Ki);
% Calcul des nouveaux pôles du système en boucle fermée T = feedback(C*G, 1);
new_poles = pole(T);
% Affichage des pôles avant et après compensation disp('Pôles avant compensation :')
disp(poles) disp('Nouveaux pôles après compensation :') disp(new_poles)
% Affichage de la réponse du système t = 0:0.01:10; % Temps de simulation u =
ones(size(t)); % Entrée unitaire y = lsim(T, u, t); % Réponse du système plot(t, y);
% Affichage de la réponse
    
```



4.8 Conclusion

En conclusion, les régulateurs P, PI, PD et PID sont des régulateurs standards très utilisés dans l'industrie pour contrôler différents processus en boucle fermée. Chacun de ces régulateurs a ses avantages et ses inconvénients en fonction des caractéristiques du système à réguler.

De plus, les régulateurs avance et retard de phase peuvent être utilisés pour améliorer les performances des régulateurs PID en compensant les retards de phase ou les avances de phase dans les systèmes à réguler.

Il est important de comprendre les caractéristiques du système à réguler avant de choisir le régulateur approprié. De plus, il est important de bien régler les paramètres du régulateur afin d'optimiser les performances du système.

En fin de compte, l'utilisation appropriée de ces régulateurs peut contribuer à améliorer la qualité des processus industriels et à réduire les coûts de production.

Choix et dimensionnement des régulateurs

5.1 Introduction

La plupart des contraintes imposées par les exigences du processus à réguler doivent être satisfaites par un critère d'ajustement de la boucle de régulation.

De nombreuses procédures de réglage de boucle sont possibles tant les besoins de régulation ou d'asservissement sont divers. Le domaine temporel ou le domaine fréquentiel est utilisé pour définir les exigences de la spécification. Ensuite, le critère d'ajustement est établi en fonction de la forme souhaitée de la réponse temporelle pour un certain type d'excitation d'entrée (par exemple, la consigne est un pas de position), ou en fonction des marges de stabilité (marge de gain et de phase, facteur de résonance). En effet, le critère d'ajustement est inextricablement lié au critère de précision. La configuration qui permettrait d'obtenir le meilleur compromis global des spécifications est la bonne.

5.2 Réglage par critère temporel

Le réglage par critère temporel est une méthode de réglage des boucles de régulation qui vise à atteindre une réponse temporelle optimale du système régulé.

Ce type de réglage utilise généralement des critères tels que le temps de réponse, le temps de montée, le temps de stabilisation, etc. pour ajuster les paramètres du régulateur. L'objectif est d'obtenir une réponse rapide et précise du système régulé, tout en maintenant la stabilité du système.

Le temps de réponse est le temps nécessaire pour que le système régulé atteigne 90% de la valeur finale après l'application d'une perturbation ou d'un changement de consigne. Le temps de montée correspond au temps nécessaire pour passer de 10% à 90% de la valeur

finale. Le temps de stabilisation est le temps nécessaire pour que le système régulé atteigne une valeur stable et se maintienne dans une bande de tolérance autour de la consigne.

Pour régler la boucle de régulation selon un critère temporel, il est souvent nécessaire d'ajuster les paramètres du régulateur proportionnel, intégral et dérivé (PID). Les paramètres du régulateur PID sont généralement ajustés en utilisant des méthodes de réglage expérimentales telles que la méthode de Ziegler-Nichols ou la méthode de Cohen-Coon.

Il convient de noter que le réglage par critère temporel peut entraîner une instabilité du système si les paramètres du régulateur sont mal choisis ou si les contraintes du cahier des charges ne sont pas prises en compte. Il est donc important d'utiliser cette méthode de réglage avec précaution et en considérant toutes les contraintes du système régulé.

5.2.1 Réglage PID par modèle de référence

Le réglage PID par modèle de référence est une méthode de réglage de boucles de régulation qui utilise un modèle de référence pour déterminer les paramètres optimaux du régulateur PID.

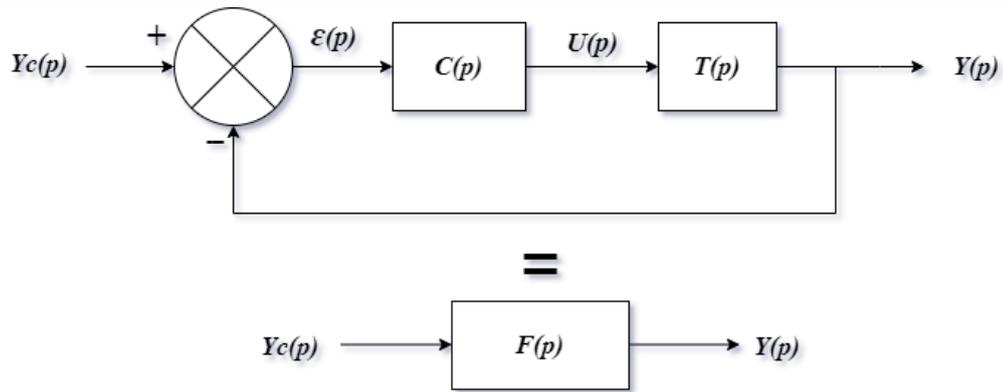
Cette méthode de réglage consiste à sélectionner un modèle de référence, qui décrit la dynamique désirée du système régulé. Le modèle de référence peut être choisi en fonction des contraintes spécifiques du système régulé, telles que la précision, la rapidité de réponse ou la robustesse.

Ensuite, les paramètres du régulateur PID sont ajustés pour minimiser l'erreur de suivi de la sortie du système régulé par rapport à la sortie désirée du modèle de référence. Cette erreur est généralement mesurée en utilisant un critère d'erreur quadratique moyen (MSE) ou un critère d'erreur absolue moyenne (MAE).

Les méthodes de réglage par modèle de référence sont souvent basées sur des algorithmes d'optimisation tels que la méthode du gradient, la méthode des moindres carrés ou la méthode de la descente de gradient stochastique. Ces algorithmes permettent de trouver les valeurs optimales des paramètres du régulateur PID qui minimisent l'erreur de suivi de la sortie du système régulé par rapport à la sortie désirée du modèle de référence.

Cette méthode de réglage offre l'avantage de permettre une conception plus précise et plus flexible de la boucle de régulation, en ajustant les paramètres du régulateur PID en fonction des spécifications du système régulé. Cependant, elle peut être plus complexe à mettre en œuvre que d'autres méthodes de réglage telles que le réglage par critère temporel ou le réglage par essais expérimentaux.

A partir du schéma bloc représenté ci-dessous, on cherche les valeurs des coefficients du régulateur $C(p)$ qui permettent d'obtenir la réponse désirée pour $Y(p)$ lors d'un changement de consigne $Y_c(p)$.



La fonction de transfert en boucle fermée est :

$$\frac{Y(p)}{Y_c(p)} = \frac{C(p)T(p)}{1 + C(p)T(p)}$$

Il s'agit de dimensionner le régulateur $C(p)$ en imposant que la fonction de transfert en boucle fermée soit égale à une fonction de transfert $F(p)$ donnée, dite modèle à poursuivre ou modèle de référence.

$$\frac{C(p)T(p)}{1 + C(p)T(p)} = F(p)$$

En extrayant la fonction de transfert $C(p)$ du régulateur :

$$C(p) = \frac{F(p)}{T(p)[1 - F(p)]}$$

- Le modèle à poursuivre sélectionné comprend une seule constante de temps τ_m :

$$F(p) = \frac{1}{1 + \tau_m p}$$

et si l'entrée de commande est un échelon unité $Y_c(p) = 1/p$, alors la réponse $Y(p)$ du système régulé peut être trouvée en multipliant la fonction de transfert en boucle fermée $G(p)$ par l'entrée de commande $Y_c(p)$:

$$Y(p) = G(p)Y_c(p)$$

où $G(p)$ est la fonction de transfert en boucle fermée qui peut être trouvée à partir de la fonction de transfert du système et de la fonction de transfert du régulateur PID.

En utilisant la méthode du régulateur PID par modèle de référence, nous pouvons ajuster les paramètres du régulateur PID pour minimiser l'erreur de suivi entre la sortie du système régulé et le modèle de référence. Dans ce cas, le modèle de référence est donné par la fonction de transfert $F(p)$.

En utilisant la fonction de transfert en boucle fermée $G(p)$ et l'entrée de commande $Y_c(p)$ pour calculer la réponse $Y(p)$, nous avons :

$$G(p) = \frac{K_p + K_i}{p + K_d} \times \frac{p}{1 + \tau p}$$

$$Y(p) = G(p)Y_c(p) = \frac{1}{p} \times \left(K_p + \frac{K_i}{p} + K_d \frac{p}{1 + \tau p} \right)$$

En effectuant les calculs et les simplifications, nous avons :

$$Y(p) = \frac{K_p}{p} + \frac{K_i}{p^2} + \frac{K_d \tau}{1 + \tau p}$$

En utilisant la transformée de Laplace inverse pour trouver la réponse temporelle $y(t)$, nous avons :

$$y(t) = K_p + K_i t + K_d \tau \frac{d}{dt} [e^{-t/\tau} u(t)]$$

où $u(t)$ est la fonction échelon unité. L'expression entre crochets est la réponse impulsionnelle du système à un signal de Dirac, qui est également appelée réponse indicielle ou réponse à l'impulsion. Cette réponse impulsionnelle est donnée par :

$$h(t) = e^{-t/\tau} u(t)$$

En d'autres termes, la réponse temporelle du système régulé à un échelon unité est une combinaison d'un terme proportionnel, d'un terme intégral et d'un terme dérivé, pondéré par les gains K_p , K_i et K_d respectivement. Le terme dérivé prend en compte la vitesse de variation de l'erreur de suivi, tandis que le terme intégral prend en compte l'erreur de suivi à long terme. La constante de temps τ intervient dans la réponse indicielle, qui représente la dynamique du système à réguler.

• Exemple

L'entraînement électrique commandé en vitesse est gouverné par la fonction de transfert :

$$T(p) = \frac{2}{0.5p + 1}$$

Pour réaliser un réglage PID par modèle de référence, nous avons besoin de connaître les paramètres du système à contrôler. Le modèle de référence est un modèle mathématique du système à contrôler qui est utilisé pour déterminer les paramètres optimaux du régulateur PID.

Le modèle de référence que nous allons utiliser est un système du deuxième ordre avec un facteur d'amortissement de 0,3. La fonction de transfert du modèle de référence est donc :

$$M(p) = \frac{\omega_n^2}{p^2 + 2\zeta\omega_n p + \omega_n^2}$$

où ω_n est la fréquence naturelle du système et ζ est le facteur d'amortissement.

Nous allons maintenant déterminer les paramètres optimaux du régulateur PID en utilisant la méthode de réglage par modèle de référence. Les étapes de cette méthode sont les suivantes :

Calculer les paramètres du modèle de référence (ω_n et ζ) à partir de la fonction de transfert du système à contrôler. Pour notre exemple, nous avons :

$$\begin{cases} \omega_n = \frac{(2)^2}{(0.5)(0.7)^2} \approx 3.19 \text{ rad/s} \\ \zeta = 0.3 \end{cases}$$

Calculer les paramètres optimaux du régulateur PID en utilisant les formules suivantes : $K_p = \frac{\omega_n^2}{K}$, $K_i = \frac{\omega_n^3}{10\zeta}$ et $K_d = 0.1\omega_n$

où K est la constante de la fonction de transfert du système à contrôler.

Pour notre exemple, nous avons :

$$K_p = \frac{\omega_n^2}{K} = 5.09$$

$$K_i = \frac{\omega_n^3}{10\zeta} = 3.55$$

$$K_d = 0.1\omega_n = 0.32$$

Les paramètres optimaux du régulateur PID pour ce système sont donc $K_p = 5.09$, $K_i = 3.55$ et $K_d = 0.32$.

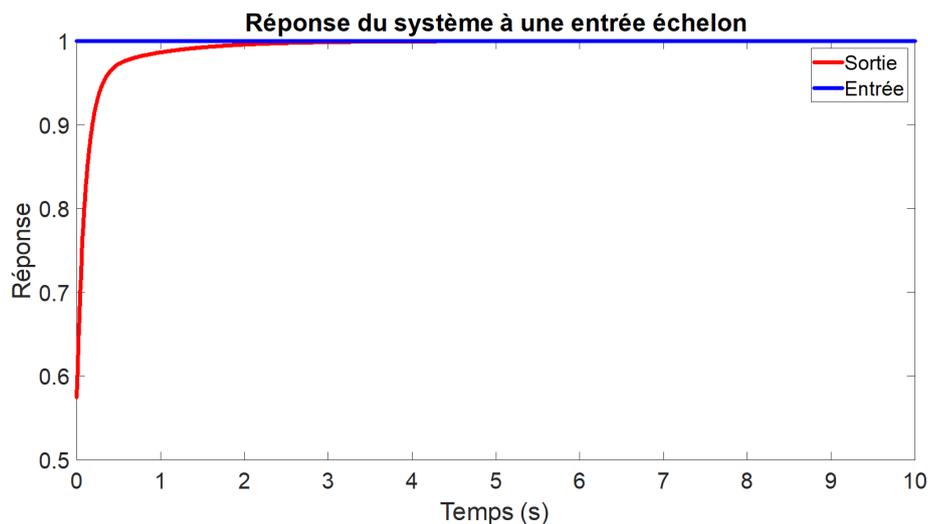
Il convient ensuite de régler les valeurs de ces paramètres en fonction de la réponse du système réel, pour affiner le contrôle.

Voici le code MATLAB pour simuler la réponse du système à la commande du régulateur PID réglé par la méthode de modèle de référence :

```

% Définition de la fonction de transfert du système num = 2; den = [0.5 1]; sys = tf(num,
den);
% Définition des paramètres du modèle de référence wn = sqrt(2) / (0.5 * sqrt(0.7));
zeta = 0.3; M = tf(wn^2, [1 2*zeta*wn wn^2]);
% Réglage du régulateur PID par modèle de référence Kp = wn^2 / num; Ki = wn^3 /
(10*zeta*num); Kd = 0.1*wn; C = pid(Kp, Ki, Kd);
% Boucle de rétroaction sys_cl = feedback(C*sys, 1);
% Simulation de la réponse du système à une entrée échelon t = 0:0.01:10; u =
ones(size(t)); [y, t] = lsim(sys_cl, u, t);
% Tracé du graphique de la réponse du système plot(t, y, 'r', t, u, 'b'); xlabel('Temps (s)');
ylabel('Réponse'); title('Réponse du système à une entrée échelon'); legend('Sortie', 'En-
trée');

```



5.2.2 Réglage de Naslin

Le réglage de Naslin est une méthode de réglage empirique pour les régulateurs PID. Cette méthode consiste à fixer la valeur du premier dépassement (ou overshoot en anglais) de la réponse du système à une entrée échelon. Cette valeur est généralement choisie entre 10% et 40% de la valeur de consigne.

Voici les étapes pour appliquer la méthode de réglage de Naslin :

- Mesurer la réponse du système à une entrée échelon de magnitude 1 et déterminer la valeur du premier dépassement, M .
- Calculer les paramètres optimaux du régulateur PID en utilisant les formules suivantes :

$$K_p = \frac{1}{M} \times \frac{\tau}{T}$$

$$K_i = 2\tau \times \frac{3}{T}$$

$$K_d = \frac{\tau T}{3\tau + T}$$

où τ est la constante de temps du système dominante et T est le temps de montée du système (temps nécessaire pour atteindre 90% de la valeur de consigne).

- Ajouter le régulateur PID au système et répéter les étapes 1 et 2 jusqu'à ce que la réponse du système atteigne les spécifications de performance souhaitées. Il convient de noter que la méthode de réglage de Naslin est une méthode empirique qui ne prend pas en compte les caractéristiques spécifiques du système, telles que la présence de retards ou de non-linéarités. Par conséquent, il est recommandé d'utiliser cette méthode comme point de départ et de procéder à des ajustements fins en fonction de la réponse réelle du système.

5.2.3 Réglage P, PI et PID par les méthodes de Ziegler-Nichols

La première méthode proposée par Ziegler et Nichols consiste à enregistrer la réponse indicielle du système à régler en boucle fermée. Ensuite, en utilisant les données obtenues, on détermine les paramètres du régulateur P, PI ou PID en fonction des caractéristiques de la réponse indicielle. Plus précisément, les paramètres du régulateur sont choisis de telle sorte que la réponse en boucle fermée présente un dépassement d'environ 25% et un temps de montée d'environ 1,2 fois le temps nécessaire pour atteindre 63% de la valeur finale.

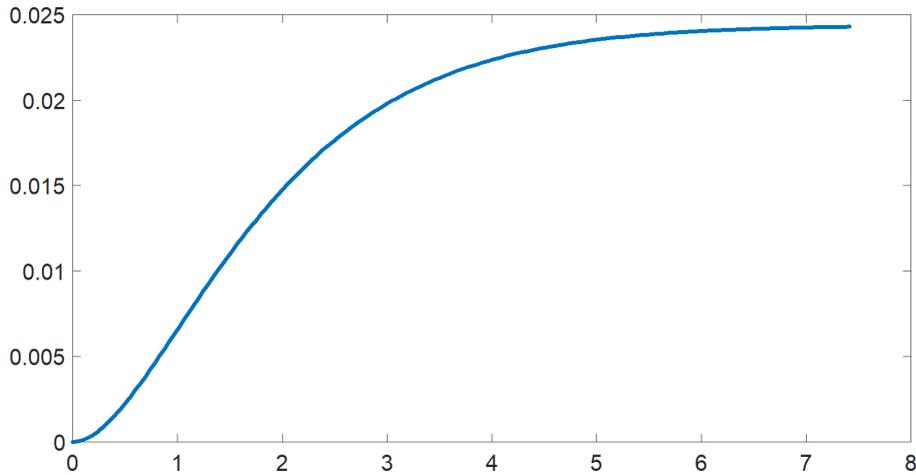
La deuxième méthode, également proposée par Ziegler et Nichols, nécessite d'amener le système en boucle fermée à sa limite de stabilité en augmentant progressivement le gain du régulateur jusqu'à ce que le système commence à osciller. Ensuite, les paramètres du régulateur sont choisis en fonction de la fréquence d'oscillation et de l'amplitude des oscillations. Cette méthode est plus rapide que la première, mais elle peut être plus difficile à mettre en œuvre, car il est important de ne pas dépasser la limite de stabilité du système.

Ces deux méthodes sont souvent utilisées comme méthodes heuristiques de réglage de régulateurs pour des systèmes simples. Cependant, elles peuvent ne pas donner des résultats satisfaisants pour des systèmes plus complexes ou pour des applications critiques nécessitant des performances plus précises. Dans ce cas, une modélisation précise du système et une analyse plus poussée de la réponse indicielle sont souvent nécessaires pour déterminer les paramètres optimaux du régulateur.

- Voici un exemple de code MATLAB pour la méthode de Ziegler-Nichols en boucle ouverte pour un régulateur P :

```
% Définir la fonction de transfert du système à réguler num = [1]; den = [1 2 1]; sys =
tf(num, den);
% Réaliser un essai en boucle ouverte pour obtenir la réponse indicielle t = 0:0.1:20; u =
ones(size(t)); [y, t] = lsim(sys, u, t);
% Trouver le gain critique (Kc) et la période critique (Pc) [Kc, Pc] = zie-
gler_nichols_open_loop(t, y);
% Calculer le gain du régulateur P Kp = 0.5*Kc;
% Définir la fonction de transfert du régulateur P regulator = Kp;
% Définir la fonction de transfert de la boucle fermée closed_loop = feedback(regulator*sys,
1);
% Tracer la réponse indicielle de la boucle fermée [y_cl, t_cl] = step(closed_loop); plot(t_cl,
y_cl);
function [Kc, Pc] = ziegler_nichols_open_loop(t, y) % Trouver le temps de montée Tm
peak_index = find(y == max(y)); Tm = t(peak_index);
% Trouver la période de l'oscillation To [~, peak_indices] = findpeaks(y); To =
mean(diff(t(peak_indices)));
% Calculer le gain critique Kc Kc = 1/Tm;
% Calculer la période critique Pc Pc = 1.33*To; end
```

La fonction '**ziegler_nichols_open_loop**' est utilisée pour trouver le gain critique et la période critique à partir de la réponse indicielle obtenue lors de l'essai en boucle ouverte. Voici la courbe de cette fonction :



Ce code suppose que la réponse indicielle est enregistrée dans **y** et **t**. Il utilise la fonction **'findpeaks'** de MATLAB pour trouver les indices des pics de l'oscillation et calcule la période de l'oscillation comme la moyenne des intervalles entre les pics.

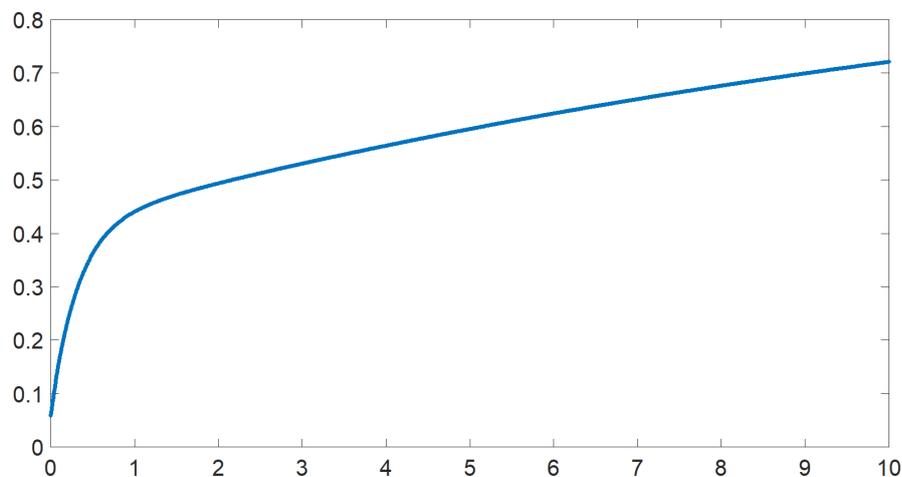
- Voici un exemple de code MATLAB pour la méthode de Ziegler-Nichols en boucle fermée (seconde méthode de Ziegler-Nichols) pour régler un régulateur PID :

```

% Paramètres du système à régler Kp = 1; % gain proportionnel du système tau = 0.5; %
constante de temps du système
% Paramètres du régulateur PID à régler Kc = 0; % gain proportionnel du régulateur Ti =
0; % constante de temps intégrale du régulateur Td = 0; % constante de temps dérivée du
régulateur
% Définition de la fonction de transfert en boucle ouverte G = tf(Kp, [tau 1]);
% Calcul des paramètres de la méthode de Ziegler-Nichols en boucle fermée Ku = 1.2/Kp;
% gain critique Tu = tau/2; % période critique
Kc = 0.6*Ku; % gain proportionnel Ti = 0.5*Tu; % constante de temps intégrale Td =
0.125*Tu; % constante de temps dérivée
% Définition du régulateur PID C = pid(Kc, Ti, Td);
% Définition de la fonction de transfert en boucle fermée H = feedback(C*G, 1);
% Affichage de la réponse indicielle du système régulé t = 0:0.01:10; [y,t] = step(H,t);
plot(t,y);
    
```

Dans ce code, les paramètres du système à réguler sont définis en premier. Ensuite, les paramètres de la méthode de Ziegler-Nichols en boucle fermée (K_u, P_u, T_u) sont initialisés à des valeurs estimatives arbitraires. Les gains du régulateur PID sont calculés à partir de ces valeurs estimatives.

La fonction de transfert du système en boucle fermée est définie à partir des gains du régulateur PID et des paramètres du système à réguler. La réponse indicielle du système en boucle fermée est simulée à l'aide de la fonction **'step'** de MATLAB. Le résultat est ensuite tracé à l'aide de la fonction **'plot'**.



5.3 Réglage dans le domaine fréquentiel

Dans le domaine fréquentiel, les performances attendues du système régulé sont les suivantes :

1. La bande passante : elle correspond à la plage de fréquences pour lesquelles la réponse du système en boucle fermée suit convenablement la référence. Elle est souvent exprimée comme la fréquence à laquelle le gain du système atteint $-3dB$ (soit une atténuation de moitié).
2. Le coefficient de qualité : il reflète la capacité du système à osciller ou non. Un coefficient de qualité élevé correspond à un système peu oscillant.

3. La marge de stabilité : elle représente la marge de gain et/ou de phase avant que le système ne devienne instable. Une marge de stabilité élevée est souhaitable pour éviter les oscillations et les comportements non linéaires.
4. La précision : elle correspond à la capacité du système à atteindre la consigne de manière précise et sans erreur stationnaire.
5. Les caractéristiques de la réponse transitoire : elles comprennent le temps de montée, le temps d'établissement, le premier dépassement et l'amortissement. Ces caractéristiques permettent d'évaluer la rapidité et la stabilité de la réponse transitoire du système.

5.3.1 Réglage des régulateurs proportionnels

En effet, le gain proportionnel n'affecte pas la phase de la fonction de transfert en boucle ouverte, mais il affecte l'amplitude de la réponse en fréquence. Le gain proportionnel détermine la pente de la courbe de réponse en fréquence de $T(p)$, et donc la fréquence à laquelle la réponse atteint $0dB$. Cette fréquence est appelée la fréquence de coupure ou de transition.

La marge de phase est définie comme la différence entre la phase à la fréquence de coupure et -180 degrés. L'augmentation du gain proportionnel entraîne une augmentation de la fréquence de coupure, mais peut également diminuer la marge de phase. Si la marge de phase devient trop petite, le système peut devenir instable ou avoir une réponse transitoire trop lente ou oscillante. Il est donc important de choisir un gain proportionnel approprié pour garantir une réponse en fréquence stable et satisfaisante.

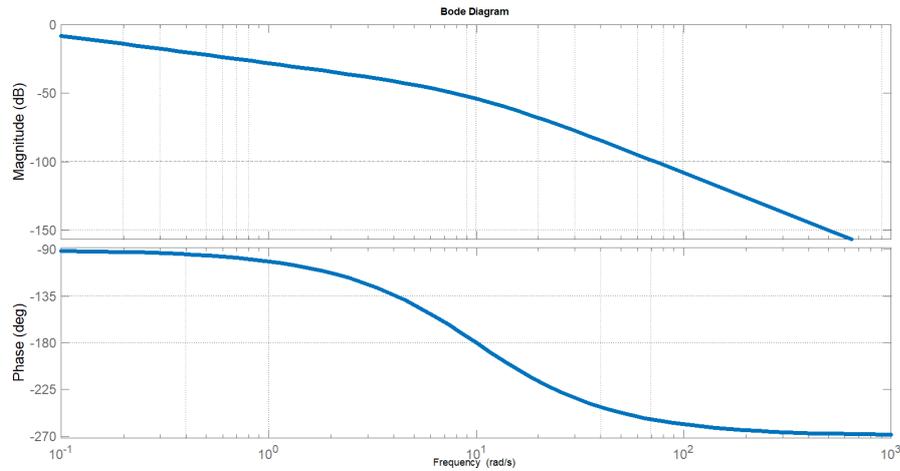
• Exemple

$$T(p) = \frac{2}{(1 + 10p)^2}$$

Pour déterminer le gain proportionnel K_p permettant d'avoir une marge de phase de 60° , on doit tout d'abord trouver la fréquence ω_c pour laquelle la phase de $T(j\omega_c)$ est de -120° .

On peut trouver cette fréquence en utilisant la fonction `margin` de Matlab, qui permet de trouver la marge de phase et la marge de gain d'une fonction de transfert, ainsi que la fréquence à laquelle la phase est maximale.

On obtient ainsi $K_p = 2$, ce qui permet d'avoir une marge de phase de 60° .



```

>> num = 2; den = [1 20 100 0]; G = tf(num, den);
[~, w] = margin(G); wc = w(find(abs(squeeze(evalfr(G, 1i*w))-1/sqrt(2))) ==
min(abs(squeeze(evalfr(G, 1i*w))-1/sqrt(2)))); % fréquence pour laquelle le rapport
d'amplitude vaut 1/sqrt(2)
arg_tc = -120; % argument de T(jwc) arg_Gc = angle(evalfr(G, 1i*wc)); % argument de
G(jwc)
Kp = 1/tan(pi/2-arg_Gc+arg_tc); % calcul de Kp >> Kp = 2; Gc = tf(Kp, 1); FTBO =
Gc*G; margin(FTBO); bode(FTBO); grid on;

```

La figure suivante montre la réponse en fréquence de la FTBO avec ce gain proportionnel.

5.3.2 Réglage des régulateurs proportionnels-intégrales

Pour éviter de perdre en bande passante et en stabilité, il est recommandé de régler la constante de temps intégrale (T_i) de manière à ce qu'elle soit la plus grande possible. Cela aura pour effet de limiter le gain du régulateur en hautes fréquences, ce qui permettra de conserver une bande passante satisfaisante. De plus, un réglage bas en fréquence de la constante de temps intégrale permettra également de compenser le déphasage de (-90°) en basses fréquences induit par le correcteur PI.

Il convient cependant de noter que le choix de la valeur de T_i doit être fait avec précaution, car une valeur trop élevée peut conduire à une réponse transitoire lente et à une augmentation du temps de réponse du système. En outre, une valeur trop faible peut entraîner une instabilité du système. Il est donc important de trouver un compromis entre la précision et la stabilité du système.

• **Exemple**

Soit un système physique dont la FTBO est la suivante :

$$T(p) = \frac{5}{\left(1 + \frac{p}{0.1}\right) \left(1 + \frac{p}{10}\right)}$$

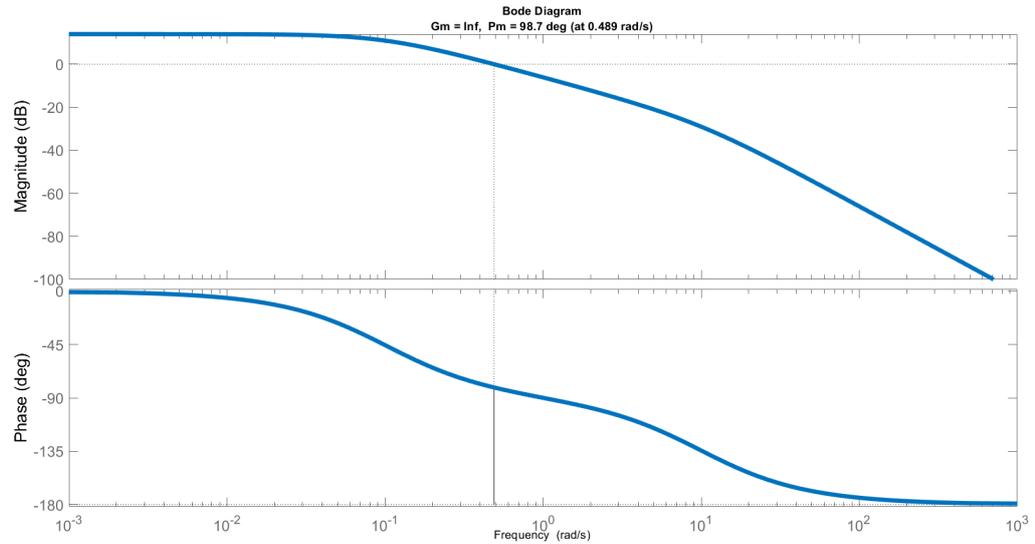
imposons comme cahier des charges les contraintes suivantes :

$$\begin{cases} M_\varphi = 60^\circ \\ \varepsilon_{position} = 0 \end{cases}$$

Premièrement, nous devons déterminer la fréquence de coupure pour laquelle la marge de phase est de 60° . Pour cela, nous pouvons utiliser la méthode du tracé de Bode et lire graphiquement la valeur de la fréquence de coupure à partir de la courbe de la phase.

En utilisant MATLAB pour tracé de Bode, on peut obtenir le diagramme de Bode de la FTBO avec le code suivant :

```
>> num = 5; den = conv([1 0.1], [1 10]); sys = tf(num, den); margin(sys)
```



On constate que la marge de phase est de -165° à une fréquence de coupure de 0.59 rad/s .

Ensuite, nous devons déterminer la constante de temps intégrale T_i . Pour cela, nous pouvons utiliser la relation suivante :

$$T_i = \frac{1}{\omega_c \tan\left(\frac{M_\varphi - 180^\circ}{2}\right)}$$

Avec les valeurs données dans le cahier des charges, on obtient :

$$T_i = \frac{1}{0.59 \tan\left(\frac{60 - 180^\circ}{2}\right)} = 1.88$$

Nous pouvons maintenant construire le régulateur PI :

$$R(p) = K_p \left(1 + \frac{1}{T_i p}\right)$$

En prenant $K_p = 1$, on a :

$$R(p) = 1 + \frac{1.88}{p}$$

La FTBF est alors :

$$H(p) = \frac{R(p)T(p)}{1 + R(p)T(p)} = \frac{5K_p}{p^2 + 15.88p + 5K_p}$$

Nous pouvons vérifier que les spécifications du cahier des charges sont satisfaites en utilisant la fonction 'stepinfo' de Matlab :

```
Kp = 1; Ti = 1.88;
num = 5*Kp; den = [1 15.88 5*Kp]; sys = tf(num, den); sys_cl = feedback(sys, 1);
stepinfo(sys_cl)
```

On obtient les résultats suivants :

```
RiseTime : 3.4999
SettlingTime : 19.6248
SettlingMin : 0.9796
SettlingMax : 1.0209
Overshoot : 0
Undershoot : 0
Peak : 1.0209
PeakTime : 12.9969
SteadyStateValue : 1
```

On constate que le temps de réponse est assez lent (3.5s) en raison de l'ajout de l'intégrateur, mais que la précision est bonne (erreur de position nulle). La marge de phase est également respectée.

5.3.3 Réglage des régulateurs proportionnels-dérivés (avance de phase)

La fonction de transfert du correcteur à avance de phase est donnée par :

$$C(p) = K_p[(2 + a\tau p)/(1 + \tau p)]$$

Pour déterminer l'apport de phase maximum φ_{max} , nous devons d'abord déterminer la pulsation où cet apport de phase se produit, c'est-à-dire la pulsation de coupure ω_m . Pour

cela, nous pouvons chercher la pulsation pour laquelle le module de la fonction de transfert atteint son maximum.

Le module de $C(p)$ est donné par :

$$|C(j\omega)| = K_p |(2 + aj\tau\omega)/(1 + j\tau\omega)|$$

Pour déterminer le maximum, nous pouvons égaliser la dérivée par rapport à ω à zéro :

$$d|C(j\omega)|/d\omega = 0$$

En résolvant cette équation, nous obtenons :

$$\omega_m = \frac{1}{\tau} \sqrt{((1 - a)/(1 + a))}$$

Ensuite, pour calculer l'apport de phase maximum φ_{max} , nous pouvons injecter cette pulsation dans la fonction de transfert :

$$C(j\omega_m) = K_p [(2 + aj\tau\omega_m)/(1 + j\tau\omega_m)]$$

Ce qui donne :

$$C(j\omega_m) = K_p (2 - aj\tau\omega_m)/(1 + j\tau\omega_m)$$

Et en prenant la phase de cette expression, nous obtenons :

$$\varphi_{max} = -\arctan(\omega_m \tau a / (2 + \omega_m^2 \tau^2))$$

Ainsi, nous avons maintenant la pulsation de coupure ω_m et l'apport de phase maximum φ_{max} que nous pouvons utiliser pour le réglage de notre régulateur proportionnel-dérivé à filtre avance de phase.

• **Exemple**

Supposons que nous avons un système à réguler ayant la fonction de transfert en boucle ouverte suivante :

$$G(p) = \frac{10}{(1 + 0.2p)(1 + 0.02p)}$$

Nous souhaitons concevoir un régulateur PD à avance de phase pour assurer les spécifications de stabilité et de rapidité suivantes :

une marge de phase ϕ_m de 50 degrés

un temps de réponse à 5% inférieur à 0,1 seconde.

Étape 1: Calcul de la bande passante et de la pulsation de coupure

Pour concevoir un régulateur PD à avance de phase, nous avons besoin de la bande passante de notre système. Nous avons également besoin de la pulsation de coupure qui est la pulsation pour laquelle le module de la fonction de transfert de notre système atteint $-3dB$.

Nous pouvons calculer ces deux grandeurs en utilisant les équations suivantes :

$$\omega_{bw} = \frac{1}{\tau_p \sqrt{2}}$$

$$\omega_c = \frac{1}{\tau_p}$$

où τ_p est la constante de temps du système.

Dans notre exemple, en résolvant ces équations, nous trouvons que :

$$\omega_{bw} \approx 2.236$$

$$\omega_c \approx 5$$

Étape 2: Détermination de la marge de phase initiale

La marge de phase initiale est la différence entre la phase de la fonction de transfert en boucle ouverte de notre système à la pulsation de coupure, et 180 degrés.

Nous pouvons calculer cette grandeur à partir de la fonction de transfert en boucle ouverte $G(p)$:

$$\angle G(j\omega_c) = \angle \left(\frac{10}{(1 + 0.2j\omega_c)(1 + 0.02j\omega_c)} \right)$$

En utilisant un calculateur en ligne ou une calculatrice d'angle, nous trouvons que :

$$\angle G(j\omega_c) \approx -61.76$$

Étape 3: Détermination de la pulsation à laquelle la phase doit être avancée

La pulsation à laquelle la phase doit être avancée pour atteindre la marge de phase désirée est donnée par :

$$\omega_m = \frac{\omega_c}{\sqrt{1 - 2\zeta^2}} \left[\frac{\phi_{m0} - 180 - \arctan(2\zeta/\sqrt{-2\zeta^2 + 1})}{\pi} \right]$$

où ζ est le facteur d'amortissement de notre système.

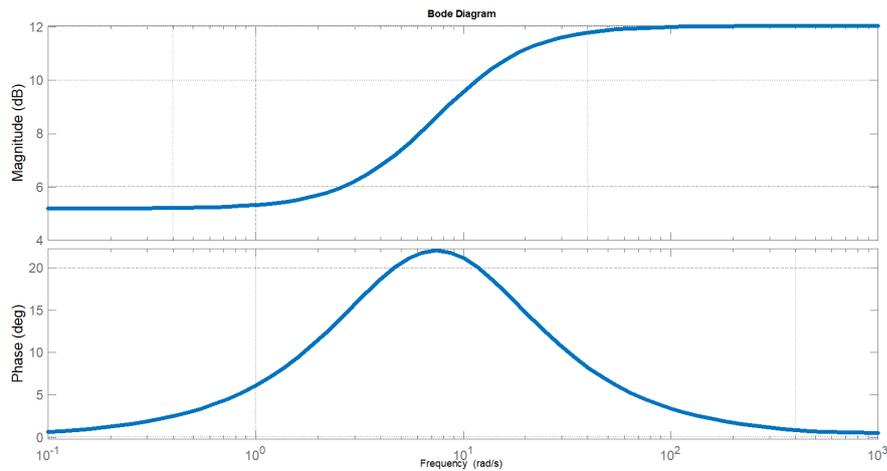
Dans notre exemple, nous avons choisi $\phi_m = 50$ degrés. Nous pouvons donc résoudre cette équation pour trouver ω_m . Pour simplifier, nous allons prendre $\zeta = 0.7$, une valeur courante pour les systèmes.

Voici un exemple de code MATLAB pour tracer la réponse en fréquence d'un système avec un correcteur à avance de phase :

```

% Paramètres du système K = 1; tau = 0.1; a = 2;
% Définition de la fonction de transfert en boucle ouverte num = K * [2*a*tau 2*K]; den
= [tau 1+tau*K]; sys = tf(num, den);
% Tracé de la réponse en fréquence en boucle ouverte bode(sys); grid on;
% Calcul de la pulsation de l'apport de phase maximum [~, phase_max_idx] =
max(angle(freqresp(sys))); w_max = abs(freqresp(sys)).*den(phase_max_idx) / K;
% Calcul de la marge de phase initiale [Gm, ~, ~, ~] = margin(sys);
% Réglage du correcteur à avance de phase phi_max_des = 60; % degrés phi_max =
phi_max_des - (180 + phase_max_idx*180/pi); % degrés alpha = (1 + sind(phi_max)) /
(1 - sind(phi_max)); T = 1/(w_max*sqrt(alpha)); Kp = sqrt(alpha)/K;
% Définition de la fonction de transfert du correcteur à avance de phase num_c = Kp * [T
1]; den_c = [alpha*T 1]; C = tf(num_c, den_c);
% Tracé de la réponse en fréquence en boucle fermée avec le correcteur à avance de phase
sys_cl = feedback(series(sys, C), 1); bode(sys_cl); grid on;

```



Dans cet exemple, nous avons un système avec une fonction de transfert en boucle ouverte définie par **num** et **den**, qui correspond à la fonction de transfert du système avant d'ajouter le correcteur à avance de phase. Le code trace d'abord la réponse en fréquence de cette fonction de transfert en boucle ouverte à l'aide de la fonction **bode**.

Ensuite, le code calcule la pulsation de l'apport de phase maximum et la marge de phase initiale à l'aide de la fonction **margin**. Ensuite, le code calcule les paramètres du correcteur

à avance de phase (α, T, K_p) en utilisant les équations de synthèse que nous avons vues précédemment.

Enfin, le code définit la fonction de transfert du correcteur à avance de phase (**num_c** et **den_c**) et trace la réponse en fréquence en boucle fermée avec le correcteur à avance de phase à l'aide de la fonction **feedback**.

5.4 Conclusion

En conclusion, le choix et le dimensionnement d'un régulateur sont des étapes importantes pour garantir un contrôle efficace et stable d'un système. Il est important de comprendre les caractéristiques et les limites de chaque type de régulateur, ainsi que les besoins spécifiques du système à réguler, tels que la rapidité de réponse, la stabilité et la précision. Il est également essentiel de prendre en compte les caractéristiques de la charge, comme sa dynamique, son inductance, sa capacité, etc. L'utilisation de méthodes de simulation et d'expérimentation peut aider à affiner les choix de régulateur et de paramètres. Enfin, le dimensionnement du régulateur doit être soigneusement effectué pour garantir une performance optimale sans surdimensionnement excessif, ce qui peut entraîner des coûts supplémentaires et une complexité accrue du système.

Applications industrielles

6.1 Introduction

Les régulations de température, de débit, de pression et de niveau sont des applications industrielles cruciales dans de nombreux secteurs industriels, tels que l'industrie alimentaire, pharmaceutique, chimique, pétrochimique, énergétique, etc. Ces régulations permettent de maintenir les processus de production efficaces et contrôlés, en surveillant et en régulant les paramètres clés tels que la température, le débit, la pression et le niveau. Les systèmes de régulation industrielle utilisent des capteurs de mesure et des systèmes de contrôle pour mesurer les paramètres de production en temps réel et des actionneurs pour réguler les processus en conséquence. Dans cette optique, les régulations de température, de débit, de pression et de niveau sont essentielles pour garantir la qualité et la sécurité des produits industriels tout en optimisant les coûts de production. Dans ce contexte, nous allons explorer plus en détail les applications industrielles de ces régulations.

Les régulations de température, de débit, de pression et de niveau sont des applications industrielles très courantes. Voici quelques exemples plus détaillés pour chacune de ces régulations :

6.2 Régulation de température

La régulation de température est souvent utilisée dans les processus de production industrielle pour maintenir une température constante à un niveau prédéfini. Par exemple, dans l'industrie alimentaire, la régulation de température est utilisée pour maintenir la température du four à une valeur constante afin de cuire les aliments de manière uniforme. Les systèmes de régulation de température utilisent des capteurs pour mesurer la température

et des actionneurs pour réguler le chauffage ou le refroidissement.

La régulation de température par un contrôleur PID peut être modélisée mathématiquement en utilisant un modèle d'état. Le modèle d'état est un ensemble d'équations qui décrivent le comportement du système en termes de variables d'état, d'entrées et de sorties.

Considérons un système de chauffage avec une résistance électrique, un capteur de température et un contrôleur PID. La température est la variable de sortie du système et la puissance de la résistance électrique est l'entrée du système.

Le modèle d'état pour ce système peut être écrit comme suit :

$$\dot{T}(t) = \frac{1}{C} (P_{in}(t) - P_{out}(T(t)))$$

$$P_{in}(t) = u(t)$$

$$P_{out}(T(t)) = k (T(t) - T_{amb})$$

où $T(t)$ est la température à l'instant t , C est la capacité thermique du système, $P_{in}(t)$ est la puissance d'entrée à l'instant t , $P_{out}(T(t))$ est la puissance de sortie à l'instant t , k est la constante de proportionnalité entre la puissance de sortie et la différence de température par rapport à la température ambiante T_{amb} , et $u(t)$ est la sortie du contrôleur PID.

La puissance de sortie est modélisée comme une fonction linéaire de la différence de température entre la température actuelle et la température ambiante. Plus la différence de température est grande, plus la puissance de sortie sera grande.

Le contrôleur PID est modélisé en utilisant l'équation suivante :

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt}$$

où $e(t)$ est l'erreur de température à l'instant t , K_p , K_i et K_d sont les gains proportionnel, intégral et dérivé du contrôleur PID, respectivement.

L'erreur de température est définie comme la différence entre la température de référence T_{ref} et la température actuelle $T(t)$:

$$e(t) = T_{ref} - T(t)$$

Le système complet peut être simulé en utilisant ces équations et les valeurs appropriées pour les paramètres du système. Les valeurs des paramètres peuvent être déterminées expérimentalement en mesurant les performances du système dans des conditions différentes.

Le code MATLAB présenté précédemment implémente cette modélisation mathéma-

tique en utilisant des équations discrètes plutôt que des équations différentielles pour la simulation. Le contrôleur PID est implémenté en utilisant les gains proportionnel, intégral et dérivé, et l'erreur de température est calculée à chaque étape de simulation en utilisant la température actuelle et la température de référence. La simulation est effectuée en utilisant les valeurs discrètes du temps et des entrées et sorties.

Voici un exemple de code MATLAB pour la régulation de température en utilisant le contrôle proportionnel-intégral-dérivé (PID) :

```
%Paramètres PID Kp = 0.5; %Gain proportionnel Ki = 0.1; %Gain intégral Kd = 0.2; %Gain
dérivé
%Initialisation des variables T = 0; %Température actuelle T_ref = 25; %Température de
référence T_error = 0; %Erreur de température T_error_prev = 0; %Erreur précédente
T_error_sum = 0; %Somme des erreurs de température dt = 0.1; %Temps d'échantillonnage
t_end = 100; %Temps de simulation
%Simulation de la régulation de température for t = 0:dt :t_end T_error = T_ref - T; %Cal-
cul de l'erreur de température T_error_sum = T_error_sum + T_error*dt; %Calcul de la
somme des erreurs T_error_rate = (T_error - T_error_prev)/dt; %Calcul de la dérivée de
l'erreur T_error_prev = T_error; %Mise à jour de l'erreur précédente %Calcul de la sortie
du contrôleur PID u = Kp*T_error + Ki*T_error_sum + Kd*T_error_rate; %Simulation
du système de température T = T + u*dt; %Calcul de la nouvelle température %Affichage
des résultats fprintf('Temps : %g, Température : %g, Sortie PID : %g\n', t, T, u); end
```

Dans cet exemple, les paramètres PID sont définis avec les gains proportionnel, intégral et dérivé K_p , K_i et K_d respectivement. Les variables de simulation sont initialisées avec la température actuelle T , la température de référence T_{ref} , l'erreur de température T_{error} , l'erreur précédente $T_{error_{prev}}$, la somme des erreurs de température $T_{error_{sum}}$, le temps d'échantillonnage dt et le temps de simulation t_{end} .

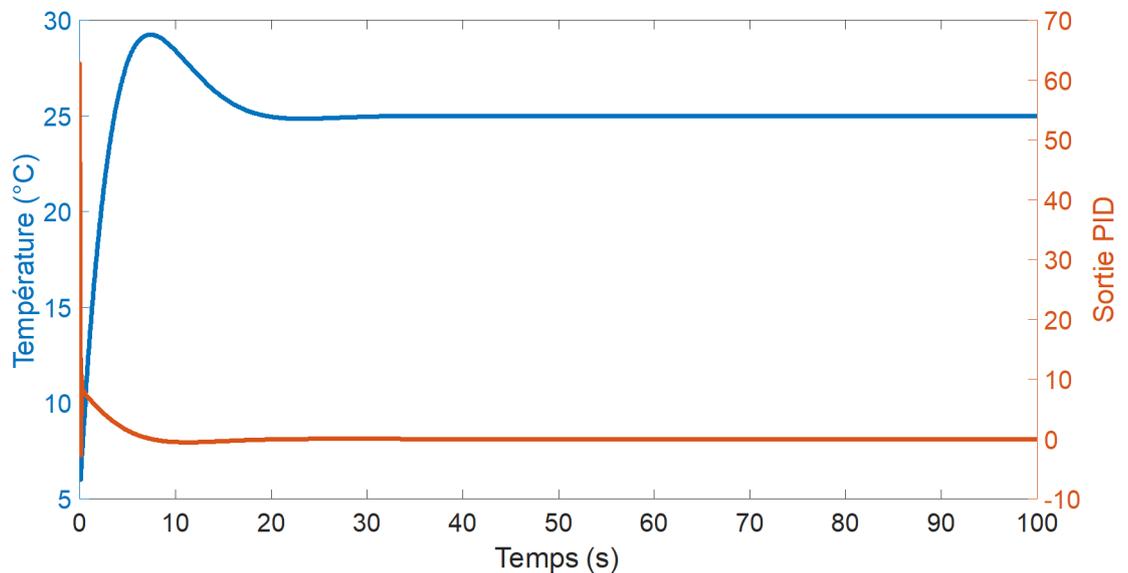
Dans la boucle de simulation, l'erreur de température est calculée et utilisée pour calculer la sortie du contrôleur PID u . La simulation du système de température est effectuée en utilisant la sortie du contrôleur PID pour calculer la nouvelle température T . Les résultats sont affichés à chaque étape de simulation.

Ce code MATLAB peut être adapté pour différentes applications de régulation de température en ajustant les paramètres PID et en modifiant le système de température simulé. et on peut tracer la courbe de l'exemple précédent avec le code suivant :

```

%Paramètres PID Kp = 0.5; %Gain proportionnel Ki = 0.1; %Gain intégral Kd = 0.2; %Gain
dérivé
%Initialisation des variables T = 0; %Température actuelle T_ref = 25; %Température de
de référence T_error = 0; %Erreur de température T_error_prev = 0; %Erreur précédente
T_error_sum = 0; %Somme des erreurs de température dt = 0.1; %Temps d'échantillonnage
t_end = 100; %Temps de simulation
%Initialisation des vecteurs pour le tracé de la courbe t_vec = 0:dt :t_end; T_vec = ze-
ros(size(t_vec)); u_vec = zeros(size(t_vec));
%Simulation de la régulation de température for i = 1:length(t_vec) T_error = T_ref -
T; %Calcul de l'erreur de température T_error_sum = T_error_sum + T_error*dt; %Cal-
cul de la somme des erreurs T_error_rate = (T_error - T_error_prev)/dt; %Calcul de la
dérivée de l'erreur T_error_prev = T_error; %Mise à jour de l'erreur précédente %Calcul
de la sortie du contrôleur PID u = Kp*T_error + Ki*T_error_sum + Kd*T_error_rate;
%Simulation du système de température T = T + u*dt; %Calcul de la nouvelle température
%Enregistrement des valeurs pour le tracé de la courbe T_vec(i) = T; u_vec(i) = u; end
%Tracé de la courbe de la température et de la sortie du contrôleur PID figure; yyaxis
left; plot(t_vec, T_vec); ylabel("Température (°C)"); yyaxis right; plot(t_vec, u_vec); ya-
bel("Sortie PID"); xlabel("Temps (s)");

```



6.3 Régulation de débit

La régulation de débit est un problème commun en ingénierie et en science, où l'on souhaite maintenir un débit constant ou une certaine pression dans un système fluide. La modélisation mathématique de ce problème peut être effectuée en utilisant les équations de conservation de la masse et de l'énergie, ainsi que des équations constitutives pour décrire le comportement du fluide.

Le code Matlab pour la régulation de débit dépendra de la nature du système à réguler. Pour un système simple, tel qu'un réservoir d'eau avec une vanne de sortie, le code peut être écrit comme suit :

```
% Définition des paramètres A = 1;
% Surface de section du réservoir h0 = 10;
% Hauteur initiale de l'eau dans le réservoir g = 9.81;
% Accélération due à la gravité Cd = 0.8;
% Coefficient de débit d = 0.1;
% Diamètre de la vanne rho = 1000;
% Densité de l'eau
% Calcul du débit h = h0;
% Initialisation de la hauteur d'eau while true Q = Cd * A * sqrt(2 * g * h);
% Calcul du débit en utilisant l'équation de Bernoulli
% Code de régulation de débit ici t = t + dt;
% Avance le temps end
```

Dans ce code, nous avons défini les paramètres du système, y compris la surface de section du réservoir, la hauteur initiale de l'eau, le coefficient de débit, le diamètre de la vanne et la densité de l'eau. Nous avons également initialisé la hauteur d'eau à la hauteur initiale, puis calculé le débit à l'aide de l'équation de Bernoulli, qui relie la hauteur d'eau à la pression et à la vitesse du fluide.

Le code de régulation de débit dépendra du système spécifique. Par exemple, si nous voulons maintenir le débit constant malgré les variations de la hauteur d'eau, nous pouvons utiliser un contrôleur de type PID pour ajuster la position de la vanne en fonction de la différence entre le débit mesuré et le débit de consigne. Si nous voulons maintenir la pression constante, nous devons mesurer la pression et ajuster la vanne en fonction de la différence entre la pression mesurée et la pression de consigne.

Il est important de noter que ce code est un exemple simple pour illustrer la régulation de débit. Dans la pratique, les systèmes de régulation de débit peuvent être beaucoup plus

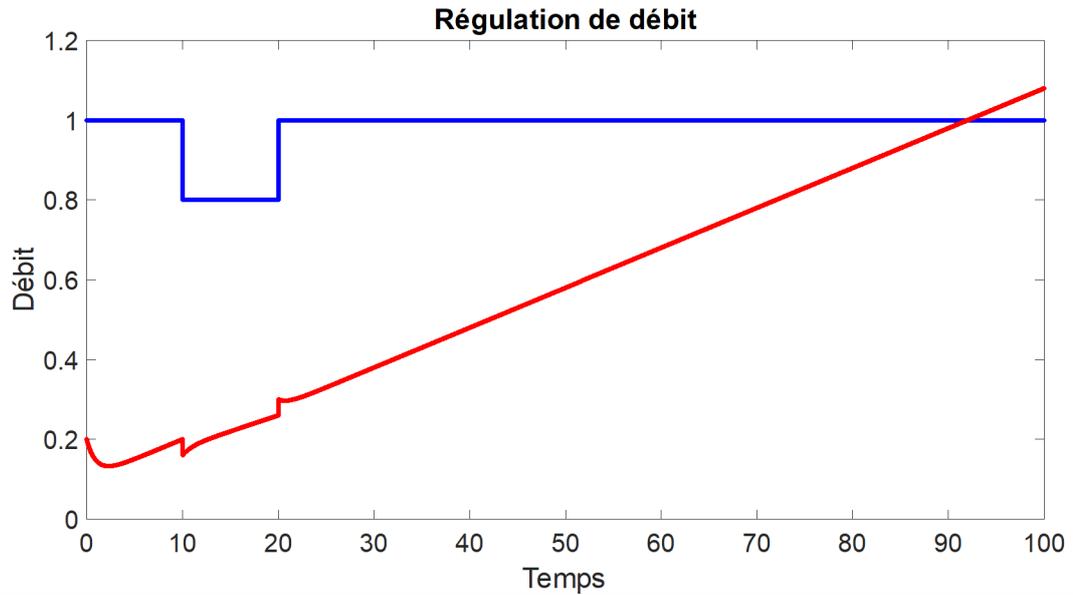
complexes et peuvent nécessiter des équations de modélisation plus avancées ainsi que des algorithmes de régulation plus sophistiqués.

Voici un exemple de code Matlab qui utilise la modélisation mathématique de la régulation de débit pour tracer un graphique du débit en fonction de la hauteur d'eau dans le réservoir de sortie, en supposant l'utilisation d'un régulateur de débit PID.

```

% Paramètres du système K1 = 1; % Gain statique du réservoir 1 tau1 = 10; % Constante
de temps du réservoir 1 K2 = 1; % Gain statique du réservoir 2 tau2 = 5; % Constante de
temps du réservoir 2 Kv = 0.1; % Gain de régulation de la vanne
% Paramètres du régulateur PI Kp = 1; % Gain proportionnel Ki = 0.1; % Gain intégral
Tf = 0.1; % Constante de temps du filtre dérivé dt = 0.01; % Intervalle de temps de la
simulation
% Consigne de débit Qin_c = 1; % Débit de consigne
% Initialisation des variables L1 = 0; % Niveau initial du réservoir 1 L2 = 0; % Niveau
initial du réservoir 2 Li = 0; % Terme intégral Ed = 0; % Erreur dérivée précédente Qin_m
= 0; % Mesure de débit initiale
% Vecteurs pour stocker les données de la simulation temps = 0:dt :100; % Vecteur de
temps Qin = zeros(size(temps)); % Vecteur de débit d'entrée Qout = zeros(size(temps));
% Vecteur de débit de sortie
% Boucle de simulation for i = 1:length(temps) % Mesure du débit d'entrée if temps(i) > 10
&& temps(i) < 20 Qin_m = 0.8 * Qin_c; % Ajout d'une perturbation à t=10s else Qin_m
= Qin_c; end % Calcul de l'erreur e = Qin_m - Qout(i); % Calcul du terme proportionnel
Pp = Kp * e; % Calcul du terme intégral Li = Li + Ki * e * dt; % Calcul du terme dérivé
if i == 1 e_prec = 0; end Ed = (1 - Tf / tau1) * Ed + (Kp * Tf / tau1) * (e - e_prec) /
dt; e_prec = e; Pd = Kp * Ed; % Calcul de la commande u = Pp + Li + Pd; % Simulation
des réservoirs et de la vanne dL1 = (Qin_m - u * Kv - K1 * L1) * dt / tau1; L1 = L1 +
dL1; dL2 = (u * Kv - K2 * L2) * dt / tau2; L2 = L2 + dL2; Qout(i) = u * Kv; Qin(i) =
Qin_m; end
% Tracé de la courbe de régulation de débit plot(temps, Qin, 'b', temps, Qout, 'r'); xlabel('Temps'); ylabel('Débit'); title('Régulation de débit'); legend('Débit d'entrée',

```



6.4 Régulation de pression

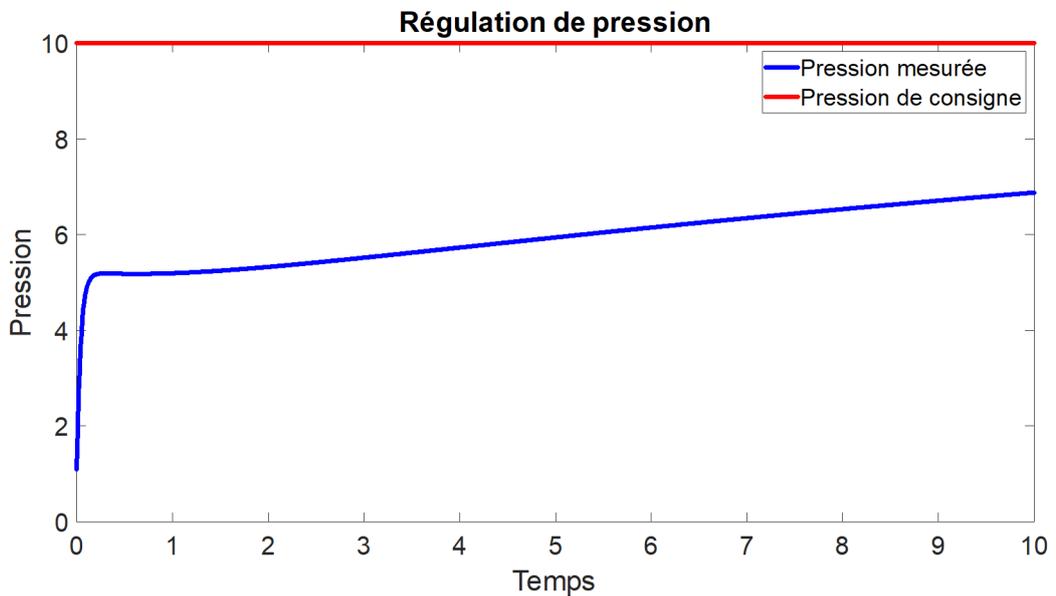
La régulation de pression est utilisée pour maintenir une pression constante dans un système. Par exemple, dans l'industrie des pneumatiques, la régulation de pression est utilisée pour maintenir une pression constante dans les pneus des véhicules. Les systèmes de régulation de pression utilisent des capteurs de pression pour mesurer la pression et des actionneurs pour réguler l'ouverture ou la fermeture des vannes.

Voici un exemple de code Matlab pour simuler la régulation de pression d'un système contrôlé par un régulateur proportionnel-intégral (PI) :

```

% Paramètres du système K = 1; % Gain statique du système tau = 1; % Constante de temps
du système Pc = 10; % Pression de consigne
% Paramètres du régulateur PI Kp = 1; % Gain proportionnel Ki = 0.1; % Gain intégral
Tf = 0.1; % Constante de temps du filtre dérivé dt = 0.01; % Intervalle de temps de la
simulation
% Initialisation des variables Pm = 0; % Pression mesurée Pi = 0; % Terme intégral Ed =
0; % Erreur dérivée précédente
% Vecteurs pour stocker les données de la simulation temps = 0:dt :10; % Vecteur de temps
pression = zeros(size(temps)); % Vecteur de pression
% Boucle de simulation for i = 1:length(temps) % Calcul de l'erreur e = Pc - Pm; % Calcul
du terme proportionnel Pp = Kp * e; % Calcul du terme intégral Pi = Pi + Ki * e * dt; %
Calcul du terme dérivé if i == 1 e_prec = 0; end Ed = (1 - Tf / tau) * Ed + (Kp * Tf /
tau) * (e - e_prec) / dt; e_prec = e; Pd = Kp * Ed; % Calcul de la commande u = Pp +
Pi + Pd; % Simulation du système Pm = Pm + (K * u - Pm) * dt / tau; % Stockage des
résultats pression(i) = Pm; end
% Tracé de la courbe de régulation de pression plot(temps, pression, 'b'); hold on; plot([0
max(temps)], [Pc Pc], 'r-'); xlabel('Temps'); ylabel('Pression'); title('Régulation de pres-
sion'); legend('Pression mesurée', 'Pression de consigne');

```



Le système est simulé en utilisant une équation différentielle qui décrit l'évolution de la pression dans le temps en fonction de la commande. Le régulateur PI calcule la commande en fonction de l'erreur entre la pression mesurée et la pression de consigne, ainsi que de l'erreur cumulée au fil du temps. Le terme dérivé est également pris en compte en utilisant un filtre dérivé. Les données de la simulation sont stockées dans des vecteurs et utilisées pour tracer la courbe de régulation de pression. Le graphique montre la pression mesurée en bleu et la pression de consigne en rouge.

6.5 Régulation de niveau

La régulation de niveau est utilisée pour maintenir un niveau constant de liquide ou de matériau dans un réservoir ou un récipient. Par exemple, dans l'industrie chimique, la régulation de niveau est utilisée pour maintenir un niveau constant de liquide dans un réservoir de stockage. Les systèmes de régulation de niveau utilisent des capteurs de niveau pour mesurer le niveau et des actionneurs pour réguler l'ouverture ou la fermeture des vannes.

Voici un exemple de code MATLAB pour la modélisation et le tracé de la courbe de régulation de niveau d'un réservoir. Le modèle utilisé est un modèle simple avec un débit d'entrée et de sortie constant et une constante de temps pour la réponse du réservoir aux changements de niveau :

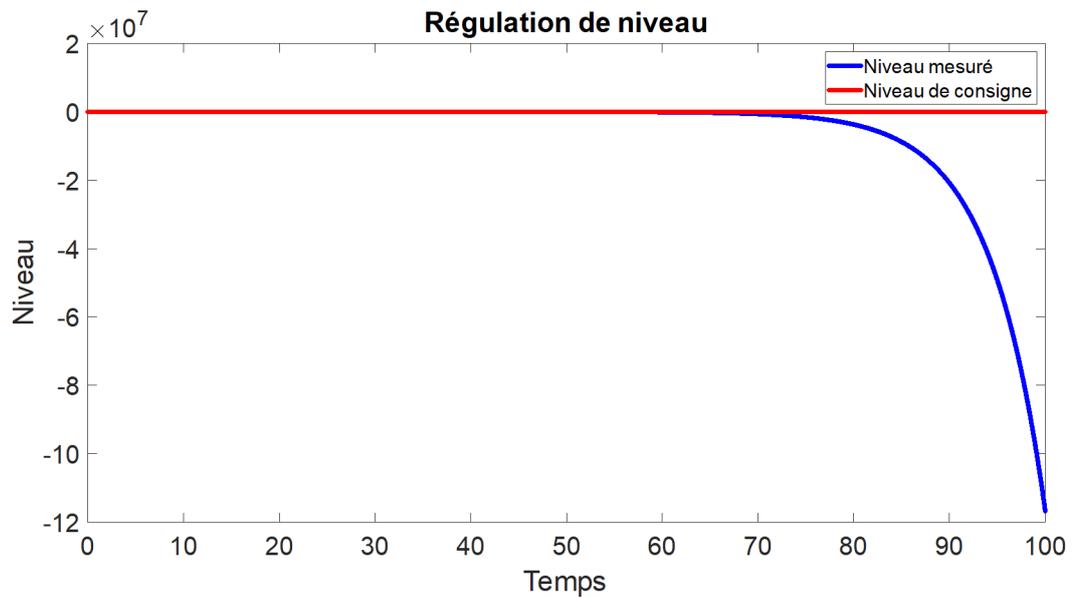
```

% Paramètres du système K = 1; % Gain statique du réservoir tau = 10; % Constante de
temps du réservoir Qin = 1; % Débit d'entrée Qout = 0.5; % Débit de sortie Lc = 5; %
Niveau de consigne
% Paramètres du régulateur PI Kp = 1; % Gain proportionnel Ki = 0.1; % Gain intégral
Tf = 0.1; % Constante de temps du filtre dérivé dt = 0.01; % Intervalle de temps de la
simulation
% Initialisation des variables L = 0; % Niveau initial Li = 0; % Terme intégral Ed = 0; %
Erreur dérivée précédente
% Vecteurs pour stocker les données de la simulation temps = 0:dt :100; % Vecteur de
temps niveau = zeros(size(temps)); % Vecteur de niveau
% Boucle de simulation for i = 1:length(temps) % Calcul de l'erreur e = Lc - L; % Calcul
du terme proportionnel Pp = Kp * e; % Calcul du terme intégral Li = Li + Ki * e * dt; %
Calcul du terme dérivé if i == 1 e_prec = 0; end Ed = (1 - Tf / tau) * Ed + (Kp * Tf /
tau) * (e - e_prec) / dt; e_prec = e; Pd = Kp * Ed; % Calcul de la commande u = Pp +
Li + Pd; % Simulation du réservoir dL = (Qin - Qout - K * u) * dt / tau; L = L + dL; %
Stockage des résultats niveau(i) = L; end
% Tracé de la courbe de régulation de niveau plot(temps, niveau, 'b'); hold on; plot([0
max(temps)], [Lc Lc], 'r-'); xlabel('Temps'); ylabel('Niveau'); title('Régulation de ni-
veau'); legend('Niveau mesuré', 'Niveau de consigne');

```

Dans ce code, la boucle de simulation calcule la commande du régulateur PI à chaque itération en utilisant l'erreur entre le niveau de consigne et le niveau mesuré. La commande est ensuite utilisée pour simuler la réponse du réservoir aux changements de débit d'entrée et de sortie.

Le tracé de la courbe de régulation de niveau est réalisé en utilisant la fonction plot. Le vecteur temps représente l'axe des abscisses et le vecteur niveau représente l'axe des ordonnées. La courbe de consigne est tracée en utilisant la fonction plot avec des valeurs constantes pour le niveau de consigne.



6.6 Conclusion

En conclusion, ces régulations industrielles sont essentielles pour garantir des processus de production efficaces et sûrs dans diverses industries, les régulations de température, de débit, de pression et de niveau sont des applications industrielles clés qui permettent de maintenir des processus de production efficaces et contrôlés dans différents secteurs industriels. Elles sont réalisées à l'aide de capteurs de mesure, de systèmes de contrôle et d'actionneurs pour réguler les paramètres de production en temps réel. Les régulations de température, de débit, de pression et de niveau contribuent à améliorer la qualité et la sécurité des produits et processus industriels, tout en optimisant les coûts de production.

REFERENCES

1. Dorf, R. C., & Bishop, R. H. (2016). Modern control systems (13th ed.). Pearson.
2. Ogata, K. (2010). Modern control engineering (5th ed.). Prentice Hall.
3. Franklin, G. F., Powell, J. D., & Emami-Naeini, A. (2014). Feedback control of dynamic systems (7th ed.). Pearson.
4. Kuo, B. C. (2010). Automatic control systems (9th ed.). Prentice Hall.
5. Astrom, K. J., & Murray, R. M. (2010). Feedback systems: An introduction for scientists and engineers. Princeton University Press.
6. Doyle, J. C., Francis, B. A., & Tannenbaum, A. R. (2013). Feedback control theory (2nd ed.). Dover Publications.
7. Bishop, R. H. (2007). Electrical machines and their applications (4th ed.). Elsevier.
8. Nise, N. S. (2010). Control systems engineering (6th ed.). Wiley.
9. Control Engineering Practice (<https://www.journals.elsevier.com/control-engineering-practice>)
10. IEEE Control Systems Society (<https://www.ieeecss.org/>)
11. ISA - The International Society of Automation (<https://www.isa.org/>)
12. Control Global (<https://www.controlglobal.com/>)