ALASCA — Architecture logicielles avancées pour les systèmes cyber-physiques autonomiques

© Jacques Malenfant

Master informatique, spécialité STL – UFR 919 Ingénierie

Sorbonne Université
Jacques.Malenfant@lip6.fr

Cours 6

Contrôle et autonomicité

Objectifs pédagogiques du cours 6

- Développer les notions fondamentales du **contrôle en boucle fermée** et comprendre à travers un exemple de l'informatique autonomique le lien entre la théorie du contrôle et l'auto-adaptabilité.
- Introduire quatre **visions** du contrôle des systèmes :
  - celle de l'*automaticien* et des systèmes le plus souvent *déterministes* et *continus*,
  - celle de l'*informaticien* et des systèmes aussi le plus souvent *déterministes* mais cependant *discrets*,
  - celle du spécialiste des *systèmes hybrides*, à la fois *discrets* et *continus* et le plus souvent déterministes, et enfin
  - celle du spécialiste de la *décision* et des systèmes hybrides également mais le plus souvent *stochastiques*.
- Comprendre ces visions de manière suffisante pour identifier le type de modèles à utiliser en pratique et pour **échanger avec les spécialistes** appropriés pour définir un modèle et une solution corrects.
- Par une vision de la théorie du contrôle, identifier les principales problématiques de la **conception** et la **mise au point** des systèmes auto-adaptables qui seront couvertes dans la suite de l'UE.

Systèmes cyber-physiques autonomiques et contrôle

- Au cœur des systèmes de contrôle cyber-physiques se trouve le contrôle en boucle fermée venant de l'automatique.
  - Inspiration fondamentale de l'informatique autonomique aussi.
- Approche *ad hoc* : quand peut-on *ignorer* les problématiques connues en contrôle ? Quand l'inertie n'entre pas en jeu *i.e.*,
  - quand les adaptations sont relativement *rares*, bien *isolées* les unes des autres, de *coût et durée faibles* et
  - guidées par des évolutions de l'entité contrôlée ou de l'environnement qui sont *lentes* et *prédictibles* en comparaison de la durée et du coût des adaptations.
- Par contre, dès que
  - les conditions évoluent *rapidement* d'une manière *imprévisible* et partiellement *indépendante* de l'exécution du système, et
  - nécessitent des adaptations dont le *coût et la durée* sont *sensibles* par rapport à leur fréquence potentielle,

  il est *crucial* de comprendre les enseignements de la théorie du contrôle pour mettre en œuvre des systèmes *viables*.

Plan

La notion de contrôle d'un système

- Puisque certaines variables en entrée d'un système peuvent être volontairement (arbitrairement) fixées, il devient possible d'exercer un *contrôle* par le *choix* des valeurs de ces variables.
- Mais pouvoir fixer volontairement les valeurs de ces variables ne dit rien sur le choix à faire.
  Un critère de choix doit être utilisé !
- En se rappelant de la théorie des systèmes vue précédemment, il s'agit conceptuellement de les choisir de manière à obtenir des sorties et des états successifs jugés *préférables*, notion restant à préciser.
- La détermination d'une *loi de contrôle* permettant d'obtenir ces sorties et états préférables est le sujet central en automatique et une partie du domaine de la décision appelée le *contrôle optimal*.
  - Loi de contrôle : fonction qui produit toutes les entrées à fournir au système pendant toute la durée de son exécution.

Contrôle : vision de l'automaticien

- Nous avons vu qu'un système continu peut être modélisé par des équations résumées par le schéma suivant :

$$\mathbf{u}(t) \longrightarrow \boxed{\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t)} \longrightarrow \mathbf{y}(t) = \mathbf{g}(\mathbf{x}(t), \mathbf{u}(t), t)$$

- Dans ce schéma, la fonction $\mathbf{u}(t)$ est une *entrée contrôlée* qui influence la valeur de la *sortie* $\mathbf{y}(t)$ *à réguler*.
  - La théorie du contrôle ayant été d'abord développée pour réguler des procédés mécaniques, électriques, chimiques, etc., son vocabulaire est souvent orienté vers ce type de procédés.
  - Pour l'appliquer en informatique autonomique, il faut les prendre toujours dans une acception assez large.
- Grande question de la théorie du contrôle : *comment déterminer* $\mathbf{u}(t)$ *pour obtenir* $\mathbf{x}(t)$ *et* $\mathbf{y}(t)$ *ayant de* ***bonnes*** *propriétés ?*

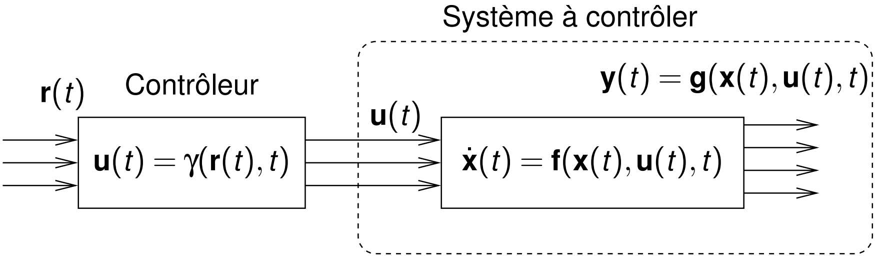
Signal de référence

- En *régulation*, le contrôle vise à maintenir la sortie mesurée au plus près d'une référence à préserver au cours du temps. *Comment exprimer cette référence ?*
- On appelle *signal de référence* une fonction $\mathbf{r}(t)$ prescrivant le comportement du système, c'est-à-dire la trajectoire que l'on souhaite lui voir suivre en fonction du temps.
- Ce signal de référence peut servir au contrôleur à calculer les valeurs des variables d'entrée $\mathbf{u}$ qui permettent d'influencer le comportement du système :

$$\mathbf{u}(t) = \gamma(\mathbf{r}(t), t)$$

  - Nota : en théorie du contrôle, γ s'appellent une *loi ou politique de contrôle* parfois aussi appelée *loi de commande*.

Schéma de contrôle obtenu



Rappel :

- $\mathbf{r}(t)$: signal de référence
- γ : loi de contrôle

Exemple : conduite automobile

Pour mieux comprendre, prenons un exemple concret :

- Considérons le pilotage d'une voiture sur un certain trajet.
- La référence $\mathbf{r}(t)$ serait alors la description (mathématique) du trajet c'est-à-dire, par exemple,
  - les positions GPS successives en fonction du temps t,
  - mais on peut aussi ajouter d'autres éléments comme le temps alloué par étape ou la vitesse maximale en tout temps t, etc.
- Si on suppose, en simplifiant, que les variables d'entrée sont :
  - α : l'angle imprimé au volant pour obtenir une direction et
  - ρ : la pression à exercer sur l'accélérateur pour obtenir une vitesse v,

  la fonction γ calculerait à tout instant t l'angle du volant $\alpha(t)$ et la pression sur l'accélérateur $\rho(t)$ permettant de suivre la trajectoire $\mathbf{y}(t)$ aussi près que possible de $\mathbf{r}(t)$ en respectant toutes ses contraintes.

La notion de rétroaction

- Bien sûr, conduire ne se résume pas à tourner le volant et appuyer sur l'accélérateur de manière prédéterminée !
- *En théorie*, certes, si tout est connu et déterministe, la position à tout instant t serait précalculable de manière sûre à partir d'une position initiale ainsi que des angles et des vitesses à tout instant.
  $\Rightarrow$ Le contrôle devient un problème d'*inversion* *i.e.*, calculer les angles et vitesses à partir d'une trajectoire souhaitée.
- Malheureusement, le système subit des pertubations : vent, état de la chaussée, piétons, autres véhicules, etc.
- Naturellement, un conducteur utilise le retour visuel (et parfois auditif ;−) qu'il perçoit sur sa trajectoire et de son environnement pour corriger sa vitesse et son angle de direction.
- *Ce retour visuel réalise une* ***rétroaction****, à savoir la connaissance de l'****état courant*** *du système, pour calculer les nouvelles valeurs des variables d'entrée*.

Boucles de contrôle

- L'utilisation du principe de rétroaction consiste à utiliser l'état $\mathbf{x}(t)$ du système au temps t dans le modèle pour calculer les nouvelles entrées, donc une loi de commande de la forme[1] :

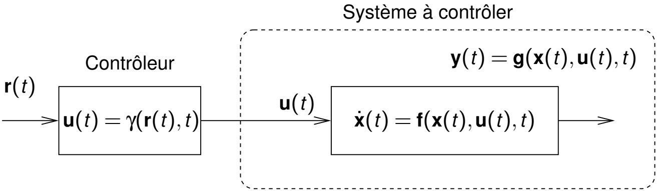
$$\mathbf{u}(t) = \gamma(\mathbf{r}(t), \mathbf{x}(t), t)$$

- Connaître l'état permet de prescrire dans $\mathbf{r}(t)$ l'état souhaité et de calculer l'écart entre l'état courant et cette référence pour produire de nouvelles entrées susceptibles de *corriger* cet écart.
- On parle alors de *contrôle en boucle fermée*, et par opposition à un contrôle *n'utilisant pas* l'état courant du système qui est appelé un *contrôle en boucle ouverte*.

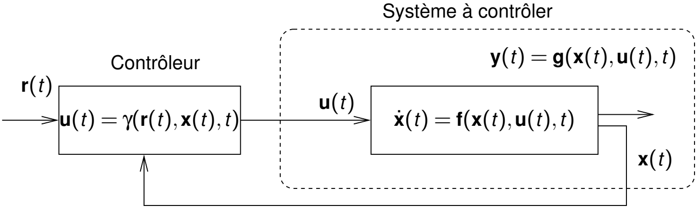
[1] Une formulation alternative utilise une boucle sur la sortie $\mathbf{y}(t)$ *i.e.*, une formulation $\mathbf{u}(t) = \gamma(\mathbf{r}(t), \mathbf{y}(t), t)$. Les deux sont donc possibles mais celle donnée ici est plus générale puisqu'on peut toujours inclure $\mathbf{y}(t)$ dans l'état $\mathbf{x}(t)$.

Schémas des boucles de contrôle ouverte et fermée

Contrôle en boucle ouverte :



Contrôle en boucle fermée :



Pourquoi une « theorie du contrôle » ?

- En principe, obtenir la loi de contrôle γ paraît relativement simple :

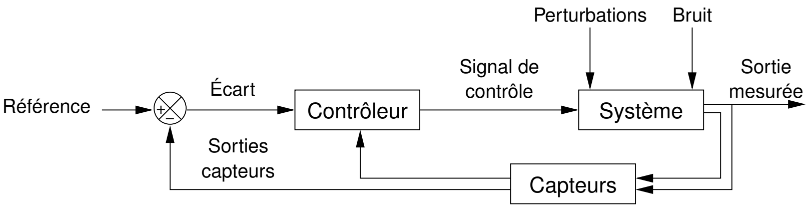
  Étant donnés une référence $\mathbf{r}(t)$ et l'état courant $\mathbf{x}(t)$, il suffit que γ calcule l'écart et connaissant les équations du système, produise les valeurs nécessaires pour corriger *tout* cet écart et se ramener *immédiatement* à la référence.
- Les problèmes soulevés par l'obtention de γ sont nombreux :
  - les équations d'états peuvent être difficiles à inverser pour obtenir les valeurs d'entrées corrigeant un certain écart ;
  - mais surtout, ce sont les *contraintes* imposées et le *temps* qui sont les sources principales de difficultés : *atteint-on les limites des actions possibles ? peut-on corriger à temps ?*
- Trouver des lois de contrôles correctes en fonction des contraintes pour des systèmes soumis à des perturbations parfois aléatoires, voilà *toute la complexité de la théorie du contrôle* !
- Bien que très actif surtout depuis environ 75 ans, le calcul des lois de contrôle optimales pour tous les types de systèmes est une thématique de recherche et d'ingénierie loin d'être épuisée. . .

Retour sur l'exemple de la conduite

- Si on considère à nouveau l'exemple de la conduite automobile, des contraintes apparaissent rapidement :
  - l'angle de direction en fonction de la vitesse (on ne braque pas à 90° à 130 km/h...),
  - la rapidité avec laquelle le véhicule peut accélérer/freiner (on ne passe pas instantanément de 130 à 50 km/h),
  - vitesse maximale à un instant donné (on ne roule pas à 250 km/h, enfin... ;-), etc.
- Exemples de contrôles *n'atteignant pas* leurs objectifs :
  1. Si l'angle de direction α est borné par $\pm\alpha^{v}_{max}$ pour une certaine vitesse v, une correction de trajectoire *souhaitée* peut devenir *impossible* si le délai de réaction est trop court.
     α de correction *insuffisant* $\Rightarrow$ *voiture dans le décor !*
  2. Si on a pris trop de retard par rapport au trajet, les limitations de vitesse peuvent empêcher de compenser ce retard, et donc la position réelle de la voiture va rester durablement en arrière de la position souhaitée par la référence.
     Vitesse de correction insuffisante $\Rightarrow$ *retard à l'arrivée !*

Plan

Schéma complet d'un système contrôlé en boucle fermée



Système :	le système contrôlé.
Perturbations :	toute influence sur la manière dont l'entrée affecte le système de manière non-contrôlée *i.e.*, paramètres externes non-contrôlés.
Bruit :	toute influence sur la qualité de la mesure de la sortie.
Sortie mesurée :	valeurs des propriétés mesurables du système.
Capteurs :	dispositifs de mesure de l'état et la sortie courants.
Référence :	valeurs souhaitées pour la sortie mesurable.
Écart :	différences entre les valeurs de référence et de la sortie capteurs.
Contrôleur :	calcule les valeurs de l'entrée contrôlée à partir des écarts.
Signal de contrôle :	valeurs de l'entrée contrôlée.

Types d'objectifs

Les objectifs d'un contrôle en boucle fermée peuvent être :

Régulation : s'assure que la trajectoire de l'état demeure autant que possible égale ou le plus près possible de la référence. Ex.: suivre la route prévue pour une voiture autonome.

Élimination des perturbations : s'assure que les perturbations n'affectent pas significativement la sortie mesurée en agissant contre ces perturbations. Ex.: pilote automatique d'un avion maintenant l'avion dans la direction souhaitée malgré le vent.

Optimisation : s'assure que la sortie mesurée est la meilleure possible étant donné un certain critère (qualité des trajectoires de l'état et de la sortie, coût des actions de contrôle) exprimé en fonction de l'état et/ou de la sortie mesurée. Ex.: faire suivre par une fusée la trajectoire de satellisation la plus économe en carburant.

Types d'approches de résolution I

- La problématique du contrôle est apparue dans différents *domaines* selon différentes *approches* et avec différentes *préoccupations*, parmi lesquelles on peut identifier :
  - L'approche de l'*automatique*, construite essentiellement autour des systèmes physiques déterministes et continus, s'est principalement intéressée à leur *stabilisation* rapide et efficace (retour à la référence sans oscillation).
  - L'approche de l'*informatique concurrente*, fondée sur les automates, s'est principalement intéressée à vérifier les propriétés comme l'*atteignabilité* des états et la capacité à amener le système dans certains états en forçant ou inhibant certaines actions de contrôle (par exemple, pour éviter l'étreinte fatale).
  - L'approche des processus markoviens commandés de la *décision*, qui s'est principalement intéressée à trouver des *lois de contrôle optimales* selon des critères donnés, en présence d'aléas dans l'évolution du système et les effets des actions de contrôle.

Types d'approches de résolution II

- L'approche des *systèmes hybrides* qui souhaite attaquer des systèmes où co-existent les propriétés des deux précédentes approches en s'attaquant explicitement aux phénomènes transitoires négatifs discret/continu engendrés par les sauts et les commutations sur la qualité des contrôles.

Et des croisements entre ces préoccupations se produisent. . .

- Les différents outils théoriques et pratiques ayant été développés dans des contextes différents, leurs hypothèses sont aussi différentes mais complémentaires :
    - Savoir parler à chacun des groupes de spécialistes en comprenant son angle d'attaque des problèmes est crucial.
    - Savoir identifier le type de système à contrôler et utiliser les bonnes approches est important.
    - Savoir quand il faut combiner certaines approches pour résoudre les problèmes plus complexes est appréciable.

Exemple : allocation dynamique de machines virtuelles

- Application web s'exécutant sur une ferme de serveurs (*cloud*) utilisant des machines virtuelles (MV).
- Aujourd'hui, on utilise deux types d'*élasticité* :

  verticale : modifier la quantité de ressources allouée aux MV pour augmenter leur capacité de traitement ;

  horizontale : modifier le nombre de MV allouées à l'application.
- Objectif : faire varier le nombre ou la taille des MV de manière à maintenir τ, le temps moyen d'exécution des requêtes (attente + service), autour d'une valeur prédéfinie τ_r (référence).
  - Si τ diminue, le coût d'opération (location des MV) augmente inutilement, et donc on peut réduire le nombre ou la taille des MV.
  - Si τ augmente, la performance se dégrade et les clients fuient, donc il faut augmenter le nombre ou la taille des MV.

Exemple : schéma de système contrôlé en boucle fermée

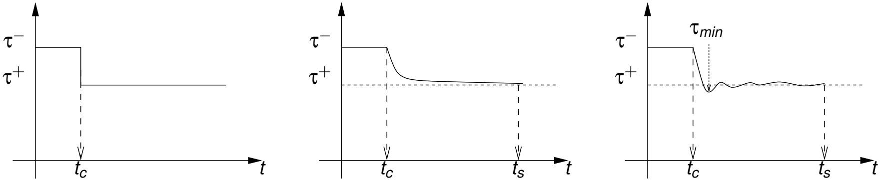
- Interprétation en termes de contrôle :
  - **Système :** application et ses MV.
  - **État et sortie mesurée :** nombre de MV courant et τ.
  - **Capteurs :** mesure et échantillonnage de τ donnant une sortie capteur $\hat{\tau}$.
  - **Référence :** τ_r.
  - **Contrôleur :** entité qui décide quand et combien de MV il faut ajouter ou retrancher en fonction de l'écart entre la sortie capteur $\hat{\tau}$ et la référence τ_r.
  - **Signal de contrôle :** séquence des ordres d'ajout ou de retrait de MV.
  - **Perturbations :** variation de τ_a, le taux d'arrivée des requêtes et de la répartition des types de requêtes venant des clients.
  - **Bruit :** erreurs de mesure de $\hat{\tau}$.

Le contrôle en pratique

- Pourquoi un contrôle comme la régulation du temps moyen d'exécution des requêtes pose-t-il tant de problèmes ?
  *Une question **essentielle** mais pas si facile à saisir !*
- Ils viennent essentiellement du fait :
  - que l'*effet* d'une adaptation n'est pas toujours *immédiat* (phénomène d'inertie du système) ;
  - que cet effet peut induire un *comportement transitoire indésirable*, comme des oscillations dans la valeur (sortie) mesurée ;
  - que l'*effet des adaptations possibles* peut être *trop puissant* ou au contraire *insuffisant* à cause des limites intrinsèques des ces adaptations ;
  - que les *perturbations* peuvent être tellement fortes qu'elles *dépassent les capacités correctives* des adaptations ;
  - qu'un *bruit trop important* rende les valeurs mesurées par les capteurs inutilisables.
- Examinons cela sur l'exemple pour mieux comprendre.

Identification du système : comportement réel

- Quel est le τ en fonction de τ_a et du nombre de MV (de la taille choisie) ? En fait M/M/1 *i.e.*, τ_a stable $\Rightarrow$ τ se stabilise.
  Exemple : on peut faire des expériences et mesurer. Supposons qu'on obtient que pour $\tau_a = 400\,r/s$, une MV maintient $\tau = 1\,s$; supposons aussi que les données expérimentales montrent que le τ à stabilisation pour un τ_a constant varie linéairement en fonction de τ_a et du nombre de MV (i.e., règle de 3).
- Comment le système réagit-il lors de l'ajout et du retrait de MV ?
  Exemple : on a 2 MV et $\tau_a = 1000\,r/s$, donc $\tau^- = 1.25\,s$; au temps t_c on ajoute une MV, à τ_a constant, donc on devrait arriver à $\tau^+ = 0.833\,s$, mais comment ?



idéal, mais physiquement irréaliste — convergence assymptotique directe — convergence assympt. oscillante

Et quel serait l'effet de la variation de τ_a pendant ce temps (perturbation) ?

Problèmes qui peuvent surgir I

Supposons qu'on choisit un délai Δ entre les adaptations d'au moins $\Delta > t_s - t_c$, i.e. $t_{i+1} = t_i + \Delta$ (sinon, instabilité...).

Scénario 1 : à chaque t_i, +1 MV si $\hat{\tau} > 1.2\,s$ et -1 MV $\hat{\tau} < 0.9\,s$.
Nota : politique avec *hystérésis*...

- État : $\tau_a = 1000\,r/s$, 2 MV, $\hat{\tau} = 1.25\,s$.
- Évolution prévisible à τ_a constant :
  - t_i : État : 2 MV, $\hat{\tau} = 1.25$, Décision : +1 MV.
  - t_{i+1} : État : 3 MV, $\hat{\tau} = 0.833$, Décision : -1 MV.
  - t_{i+2} : État : 2 MV, $\hat{\tau} = 1.25$, Décision : +1 MV.
  - t_{i+3} : État : 3 MV, $\hat{\tau} = 0.833$, Décision : -1 MV.
  - ...
- $\hat{\tau}$ *oscille entre 0.833 s et 1.25 s en ajoutant ou retranchant une MV à chaque instant de contrôle*
  $\Rightarrow$ *loi instable, à actions trop puissantes !*
- Et le client paie à chaque fois le coût fixe d'allocation...

Problèmes qui peuvent surgir II

Et si les actions étaient moins puissantes ?

Scénario 2 : idem 1 sauf que $\tau_a = 5.000\, r/s$ et 10 MV donc toujours avec un $\hat{\tau} = 1.25\, s$.

- Effet attendu de +1 MV sur $\hat{\tau}$: $\frac{10}{11} \times 1.25 = 1.14\, s$.
- Évolution prévisible à τ_a constant :
  - t_i : État : 10 MV, $\hat{\tau} = 1.25$, Décision : +1 MV.
  - t_{i+1} : État : 11 MV, $\hat{\tau} = 1.14$, Décision : ne rien faire.
  - t_{i+2} : État : 11 MV, $\hat{\tau} = 1.14$, Décision : ne rien faire.
  - ...
- $\hat{\tau}$ *se stabilise à 1.14 s, assez loin (+14%) de l'objectif de 1 s, or si on ajoutait une autre MV on obtiendrait à* t_{i+2} *12 MV donc*
  $\hat{\tau} = \frac{11}{12} \times 1.14 = 1.045$
  $\Rightarrow$ *loi trop imprécise !*

Problèmes qui peuvent surgir III

Tentons d'être plus précis encore avec un objectif très fin et des actions encore moins puissantes...

Scénario 3 : idem 2 sauf que $\tau_a = 50.000\, r/s$ et +1 MV tant que $\hat{\tau} > 1.02\, s$ et -1 MV tant que $\hat{\tau} < 0.98\, s$.

- Évolution prévisible à τ_a constant :
  - t_i : État : 100 MV, $\hat{\tau} = 1.25$, Décision : +1 MV.
  - t_{i+1} : État : 101 MV, $\hat{\tau} = \frac{100}{101} \times 1.25 = 1.237$, Décision : +1 MV.
  - t_{i+2} : État : 102 MV, $\hat{\tau} = \frac{101}{102} \times 1.237 = 1.225$, Décision : +1 MV.
  - ...
  - t_{i+23} : État : 123 MV, $\hat{\tau} = \frac{121}{122} \times 1.025 = 1.016$, Décision : ne rien faire.
  - t_{i+24} : État : 124 MV, $\hat{\tau} = 1.016$, Décision : ne rien faire.
  - ...
- *23 étapes avant d'arriver à l'objectif $\Rightarrow$ loi trop lente !*
  - Combien de clients perdus (*i.e.*, coût de l'écart à la cible) durant la convergence) ?

Problèmes qui peuvent surgir IV

Tentons d'améliorer les choses avec des actions plus progressives...

Scénario 4 : $\tau_a = 4800\, r/s$, 10 MV, donc $\hat{\tau} = 1.2\, s$.

$$\text{décision} = \begin{cases} \pm 5MV & \text{si } \hat{\tau} > 1.15\, s, \text{resp. } \hat{\tau} < 0.8\, s \\ \pm 2MV & \text{si } 1.15\, s \geq \hat{\tau} > 1.1\, s, \text{resp. } 0.8\, s \leq \hat{\tau} < 0.9\, s \\ \pm 1MV & \text{si } 1.1\, s \geq \hat{\tau} \geq 1.05\, s, \text{resp. } 0.9\, s \leq \hat{\tau} \leq 0.95\, s \\ 0 & \text{si } 1.05\, s > \hat{\tau} > 0.95 \end{cases}$$

- Évolution prévisible à τ_a constant :
  - t_i : État : 10 MV, $\hat{\tau} = 1.2$, Décision : +5 MV.
  - t_{i+1} : État : 15 MV, $\hat{\tau} = \frac{10}{15} \times 1.2 = 0.8$, Décision : -2 MV.
  - t_{i+2} : État : 13 MV, $\hat{\tau} = \frac{15}{13} \times 0.8 \approx 0.923$, Décision : -1 MV.
  - t_{i+3} : État : 12 MV, $\hat{\tau} = \frac{13}{12} \times 0.923 = 1$, Décision : ne rien faire.
- *La cible est bien atteinte, mais en acquérant 5 MV dont 3 vont devoir être recédées tout de suite*

  $\Rightarrow$ *loi à dépassement trop important !*

Qualités d'un contrôle en boucle fermée

Stabilité : (2 définitions) pour toute entrée contrôlée bornée,

1. l'état et la sortie sont également bornés ; ou
2. l'écart de l'état ou de la sortie avec la référence tend vers 0 avec le temps.

Précision : à quel point l'état demeure proche de la référence, ou plus généralement à quel point les objectifs du contrôle sont atteints.

Temps de réglage court : temps de convergence vers la référence le plus court possible après un écart entre l'état et cette dernière.

Minimum de dépassement : l'écart maximal par rapport à la référence est le plus petit possible.

*En anglais, propriétés SASO pour « **S**tability », « **A**ccuracy », « **S**hort settling time » et « small **O**vershoot ».*

Petit coup d'oeil sur la suite du cours... I

- Les difficultés induites par le **coût des mesures** et les **erreurs** dont elles peuvent être entachées sur l'état seront discutées dans au cours 7.
- Dans cet exemple, l'ajout d'une machine virtuelle peut demander de la créer, d'où un coût et un délai de mise à disposition ; plus généralement les difficultés induites par les **actionneurs** seront discutées un peu aujourd'hui et mais aussi dans les cours 8.
- Une fois qu'une MV a été ajoutée, il faudra du temps avant que le temps moyen d'exécution des requêtes se stabilise à son nouveau niveau (à taux d'arrivées des requêtes constant) ; ces difficultés liées à l'**inertie du système** sont discutées dans le reste du cours d'aujourd'hui et aux cours 11 et 13.

Petit coup d'oeil sur la suite du cours... II

- Le choix du nombre de MV à ajouter ou retrancher doit être fait judicieusement pour éviter d'augmenter le coût d'opération : c'est la politique de contrôle ; les difficultés liées au calcul d'une **bonne politique de contrôle** seront couvertes dans le reste du cours aujourd'hui et au cours 13. Il en va de même pour l'effet de la variation des perturbations, comme le taux d'arrivée des requêtes dans notre exemple.
- La théorie du contrôle examine un système à partir d'un modèle, mais en pratique il faut implanter ce modèle sur le système réel avec des entités logicielles qu'on voudra développer de manière efficace et réutiliser dans toute la mesure du possible ; les difficultés liées au **développement** et au **génie logiciel** appliqués aux entités logicielles seront couvertes dans les cours 7 à 11.

Petit coup d'oeil sur la suite du cours... III

- Une ferme de serveurs exécute rarement une seule application, or l'allocation de ses ressources aux différentes applications nécessitent une **coordination** entre chacune ; les difficultés liées à la coordination entre les entités autonomiques seront couvertes au cours 12, et leur impact sur le développement des entités seront introduites aux cours 8, 10 et 11.
- Si le système évolue ou ses perturbations et bruits changent, sa loin de commande devra aussi évoluer, ce qui donne un contrôle **adaptatif** qui doit lui-même être régulé ; cet aspect de contrôle du contrôle ou **méta-contrôle** sera discuté au cours 13.
- Le **passage à l'échelle** sera abordé en coordination au cours 12 et plus généralement au cours 14.

Plan

Origine de l'automatique

- L'automatique est née du besoin de contrôler des phénomènes essentiellement *analogiques* (continus, non numérisés) à l'aide de dispositifs eux-même *analogiques*.
- Les techniques employées consistaient à mesurer l'écart analogiquement, par un voltage par exemple, puis d'amplifier et transformer ce voltage pour impulser une action sur un actionneur.
- Exemples :
  - Artillerie anti-aérienne : régler la direction de tir (vérins, moteurs) en fonction de la position radar de la cible.
  - Missile balistique : suivre une trajectoire en agissant sur l'angle du jet de gaz du moteur et des gouvernes aérodynamiques des ailerons.

Naissance d'une discipline

- C'est l'*inertie* dans le temps propre à ces phénomènes, liées aux limites physiques intrinsèques des actionneurs (souvent mécaniques, *in fine*), qui a justifié une grande part des travaux en automatique (ou théorie du contrôle).
- Sachant qu'il s'agit d'influer sur une propriété mesurée à partir d'une propriété contrôlée, la grande question est toujours :

  > *Comment obtenir une « trajectoire » suivant une référence pour un phénomène évoluant selon sa dynamique propre (trajectoire intrinsèque) en lui appliquant une correction en continu ?*

  et subsidiairement, comment obtenir une trajectoire de *bonne qualité* (*i.e.*, SASO) ou optimale ?
- Pour attaquer formellement ce problème, les automaticiens se sont d'abord appliqués à comprendre les signaux et les effets de leur *combinaison*.

Phénomènes vus comme des signaux

- Les phénomènes étudiés en automatique sont des propriétés dont la valeur évolue dans le temps.
- Soit une propriété p, qui prend à tout instant t une certaine valeur, il est naturel de voir p comme une fonction du temps $p(t)$.
  $\Rightarrow$ En automatique, on appelle cela un *signal*.
- Contrôler consiste à appliquer un signal de contrôle, noté bien sûr $u(t)$, au signal $p(t)$ représentant la propriété à contrôler.
- L'effet de $u(t)$ est de se *combiner* à $p(t)$ de manière à en changer l'évolution, ce qui peut être capturé par une *équation de récurrence* dont la forme générale est :

$$p(t) = f(p(t), u(t))$$

- Avant même de se soucier du problème consistant à déterminer quel contrôle $u(t)$ appliquer à un signal $p(t)$ pour atteindre un certain objectif, l'automaticien étudie les propriétés de base des combinaisons $f(\cdot, \cdot)$ de signaux $p(t)$ et $u(t)$ connus.

Quelles combinaisons de signaux étudier ?

- Lorsqu'on combine des valeurs de signaux, les effets peuvent être de plusieurs ordres : additifs (ou linéaires), multiplicatifs (quadratiques, etc.), ou plus généralement non-linéaires.
- L'étude théorique des propriétés des combinaisons de signaux est un domaine très complexe en général, donc pour des raisons pratiques tenant à la fois aux possibilités de la théorie et à la réalité des applications de l'automatique, une grande part des études ont porté sur les systèmes *linéaires* où :

$$p(t) = f(p(t), u(t)) = ap(t) + bu(t)$$

- Pour des raisons historiques également, vu les premières applications, l'étude s'est aussi concentrée sur des systèmes *déterministes*, c'est-à-dire où les erreurs et autres effets aléatoires sont soit *inexistants*, soit *négligeables* (au sens où ls pouvent être négligés sans trop de problème).

Le prototype de l'automatique linéaire déterministe

- L'équation $p(t) = ap(t) + bu(t)$ est une *combinaison linéaire récurrente* dont il faut trouver une solution *i.e.*, en théorie deux alternatives :
  - Étant donné $u(t)$, quels a, b tel que $p(t) = r(t)$?
  - Étant donnés a, b, quel $u(t)$ tel que $p(t) = r(t)$?

  et les deux vont nous intéresser.
- Dans le cas où le contrôle est effectué par des contrôleurs numériques, les interventions sur le système ne peuvent se faire qu'à des instants discrets, ce qui justifie d'étudier la version discrète de l'équation :

$$p(t_{k+1}) = ap(t_k) + bu(t_k)$$

ou de manière plus simple ($t_{k+1} - t_k$ étant constant) :

$$p(k+1) = ap(k) + bu(k)$$

D'ailleurs, on peut montrer qu'il vaut mieux discrétiser l'équation avant sa résolution que discrétiser une solution continue.

Identification du système

- Le premier problème consiste à *identifier* le système c'est-à-dire obtenir les valeurs des paramètres faisant coller le modèle général au système.

  $\Rightarrow$ Dans notre exemple, les valeurs de a et b.
  - Pour certains systèmes (électricité, mécanique, etc.), il arrive que les lois connues de la physique permettent de déterminer les équations du modèle.
  - Pour de nombreux systèmes, cette approche analytique est impossible car la nature du système est inconnue ou trop complexe.
- On peut alors faire un modèle *boîte noire* avec des méthodes d'approximation statistiques comme la régression linéaire ou les séries chronologiques à partir de *données expérimentales* mesurées sur un exemplaire du système.
- Par exemple, après expérimentation où $u(k), k \geq 0$ est fixée, on peut obtenir des valeurs de a et b par régression linéaire :

$$p(k+1) = 0.43p(k) + 0.47u(k)$$

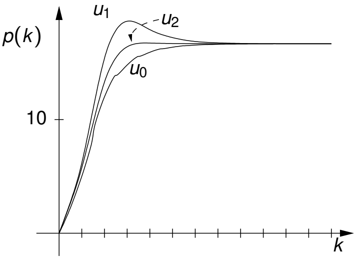
Comment réagit ce signal à un certain contrôle ?

- À partir du modèle (équation et paramètres a et b) précédent, on peut étudier le comportement de $p(t)$ pour différents $u(t)$ connus.
- Signaux de contrôle communs :
  - impulsion : $u(0) = U$, $u(k) = 0, k \geq 1$
  - saut : $u(0) = U_0$, $u(k) = U, k \geq 1$
  - linéaire : $u(k) = kU, k \geq 0$
- Appliquons à notre modèle trois signaux simples avec comme condition initiale $p(0) = 0$ et voyons les trajectoires obtenues :

$$u_0(k) = \begin{cases} 10 & k = 0 \\ 20 & k \geq 1 \end{cases}$$

$$u_1(k) = \begin{cases} 10 & k = 0 \\ 30 & k = 1 \\ 25 & k = 2 \\ 20 & k \geq 3 \end{cases}$$

$$u_2(k) = \begin{cases} 10 & k = 0 \\ 25 & k = 1 \\ 22 & k = 2 \\ 20 & k \geq 3 \end{cases}$$



Analyse des propriétés du modèle

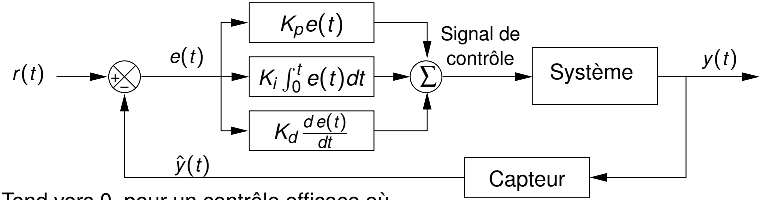
- L'automatique, puisant dans les techniques mathématiques comme les transformées de Laplace et les transformées en Z, a étudié avec beaucoup de soin les propriétés des modèles de combinaison de signaux.
- Selon les propriétés des équations, on peut arriver à conclure des choses comme, pour un $u(k)$ borné ou non :
  - le signal $p(k)$ est stable ou instable ;
  - le signal $p(k)$ oscille ou non, et s'il oscille, la fréquence de ces oscillations ;
  - le signal $p(k)$ converge à ε près en moins de n étapes.
- Et donc d'analyser dans ces conditions les propriétés SASO de ce système.

De l'analyse au choix du contrôle $u(t)$

- Dans les analyses précédentes, on s'intéresse aux propriétés générales du signal de sortie pour tout signal de contrôle.
- Ce petit énoncé introductif néglige beaucoup de choses, dont les effets des perturbations et du bruit : *on y reviendra !*
- La question brûlante est comment choisir le contrôle $u(k)$?
  - en régulation, il faut réduire l'écart entre l'état $p(k)$ et la référence $r(k)$, il est donc raisonnable de construire $u(k)$ comme fonction de $e(k)$, l'écart $p(k) - r(k)$ vu comme un signal ;
  - en élimination des perturbations, c'est le signal de perturbations qu'on veut éliminer et donc on peut construire le contrôle en fonction de ce que l'on connait de lui, déjà plus compliqué... ;
  - en optimisation, c'est encore plus complexe car on va choisir un signal de contrôle qui va minimiser une certaine fonction de coût C, et généralement le coût total qu'elle induit $\sum_0^N C(p(k), u(k))$.

Un contrôleur linéaire très courant : PID

- Considérons un contrôle en régulation où $u(k+1) = g(e(k))$ ($e(k)$ étant l'écart mesuré). Mais *quelle* fonction g ?
- Encore une fois, la simplicité et l'efficacité pratique des approches linéaires ont amené à étudier des fonctions g linéaires.
- Poser $u(k+1) = K_p e(k)$ (*proportionnel*, P), donne des résultats décevants car sans mémoire (évolution récente, dérive globale).
- Une famille de contrôleurs qui se sont montrés efficaces en pratique, qui sont donc beaucoup utilisés, est celle des contrôleurs *PID*, pour *proportionnel*, *intégrale*[1] et *dérivée*.



[1] Tend vers 0, pour un contrôle efficace où l'écart se répartit en moyenne également autour de 0.

Détermination des constantes de proportionalité I

- En discrétisant, on obtient un contrôle de la forme :

$$u(k+1) = K_p e(k) + K_i \sum_{j=0}^{k} e(j) + K_d \left(e(k) - e(k-1)\right)$$

  - il reste à déterminer les constantes K_p, K_i et K_d
- Encore une fois, les outils mathématiques développés en automatique permettent de dégrossir les choses : en régulation, un contrôleur PID réussira à éliminer l'écart, mais comment $y(t)$ va-t-elle évoluer ?
- L'automaticien va déterminer ses critères de sélection :
  - précision et temps maximal de réglage autorisé,
  - oscillations admissibles ou non,
  - maximum de dépassement, etc.

Détermination des constantes de proportionalité II

- En pratique, l'ingénieur automaticien va souvent tracer les courbes des équations obtenues avec différents paramètres K_p, K_i et K_d (en utilisant un outil comme Matlab, Mathematica, Scilab, Octave, etc.), puis examiner les résultats pour choisir le jeu de ces paramètres dont la courbe satisfera au mieux ses critères de sélection.
- On peut aussi obtenir un contrôleur PID par optimisation :
  - On définit la fonction de coût dépendant de $u(k)$ et $p(k)$, plus éventuellement d'autres paramètres internes.
  - On optimise la fonction de coût pour trouver les paramètres K_p, K_i et K_d tels qu'elle est optimale.

Perturbation et bruit

- Comme nous l'avons fait remarquer, les équations précédentes ne font pas apparaître les notions de perturbation et de bruit.
  - Cela revient à considérer que leurs effets, plus ou moins constants, plus ou moins négligeables, peuvent être complètement capturés par les paramètres de l'équation obtenus après modélisation statistique sur données expérimentales.
  - Il est possible de faire les choses de manière plus explicite.
- Pour le bruit, et les autres problèmes de mesure :
  - on peut modéliser la transformation de $y(t)$ par l'application d'une combinaison avec un signal de bruit explicite ;
  - en postulant une forme à ce signal, on peut l'entrer dans le modèle et en tenir compte pour le calcul du contrôle $u(t)$.
- Pour la perturbation :
  - on peut également introduire dans les équations le signal de perturbation attendu et en tenir compte dans le calcul du contrôle ;
  - il existe également des modèles de contrôle *prédictif* qui tiennent compte d'une prédiction du prochain niveau de la perturbation en fonction des mesures précédentes.

Retour à la modélisation à base d'états I

- La forme de modélisation présentée précédemment est axée sur le principe de *fonction de transfert*.
- Dans cette forme de modélisation, le système n'est vu que comme une entité transformant un signal de contrôle en entrée vers un signal mesuré en sortie, laissant sa dynamique interne masquée à l'intérieur d'une boîte noire (sans état explicite).
- Dans nos descriptions précédentes des systèmes, des modèles plus explicites définissent plus finement la dynamique du système au prix d'une plus grande complexité, c'est-à-dire des équations de récurrence incluant l'état x :

$$x(k+1) = f(x(k), u(k), k) \qquad x(k_0) = x_0$$
$$y(k+1) = g(x(k), u(k), k)$$

Retour à la modélisation à base d'états II

- Parce qu'il est souvent difficile de les expliciter dans leurs applications concrètes, les automaticiens se sont moins intéressés aux modèles à base d'états qu'aux modèles fondés sur les fonctions de transfert simples.
- En informatique cyber-physique et autonomique, l'accès à une description des entités contrôlés rend au contraire plus attractifs les modèles à base d'états, mais qui exigent des contrôleurs plus complexes à mettre au point.

Propriétés étudiées dans les modèles à états

- À partir du moment où la trajectoire de l'état est modélisée explicitement (et pas seulement celle de la sortie mesurée), il devient intéressant de regarder les propriétés de la dynamique du système entre les contrôles et l'espace d'états.
- Deux propriétés sont principalement considérées :

Contrôlabilité : un système est *contrôlable* si pour tout état final réalisable x_f et tout état x, il existe une séquence de contrôles $[u(0), \dots u(M)]$ qui, appliquée dans l'état $x(0) = x$, va mener à l'état $x(M) = x_f$.

Observabilité : un système est *observable* si toutes ses valeurs d'états peuvent être déduites uniquement à partir des propriétés mesurées (sorties capteurs, information partielle).

Plan

Contrôle des systèmes hybrides

- Étant continu et discret, le contrôle d'un système hybride s'exerce sur les trajectoires continues et lors des sauts discrets :
  - les trajectoires continues sont modélisées par des équations où les entrées $u(t)$ sont vues comme des signaux de contrôle continu (comme on l'a vu) ;
  - lors des sauts, l'état post-transition peut être calculé par une fonction dépendant de $v(t)$ *i.e.*, des entrées ponctuelles contrôlées influant sur l'état atteint après un saut.
- Comme pour les systèmes continus, le contrôle des trajectoires continues par un ordinateur numérique par essence discret impose une discrétisation du contrôleur continu.
- Toutefois, la discrétisation engendre des problèmes spécifiques :
  - la détection précise des instants où les sauts discrets doivent se produire et changer la loi de commande continue ;
  - les instabilités et autres propriétés SASO du contrôle continu plus difficiles à garantir lors du passage d'une trajectoire continue à la suivante par sauts discrets, car les signaux $v(t)$ et $u(t)$ combinent leurs effets aux points de sauts.

Modélisation du contrôle dans les automates hybrides

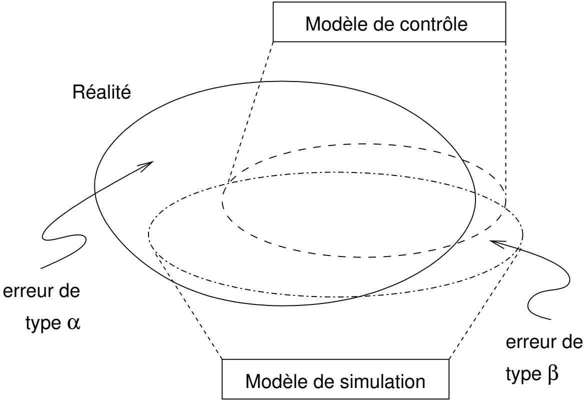
- Les systèmes hybrides sont des instruments de modélisation puissants, permettant tout autant de modéliser un système contrôlé que de modéliser un contrôleur.
- L'utilisation d'automates hybrides et de leur composition permet de mettre en place un contrôle selon deux approches :
  - monolithique : fondre la loi de contrôle directement dans le modèle du système contrôlé complet (exemple thermostat où les sauts sont provoqués par les seuils de contrôle) ;
  - modulaire : composer un système hybride contrôleur émettant les actions de contrôle en sortie et d'un système hybride contrôlé (*plant*) recevant ces actions en entrée.
- L'approche modulaire ouvre la possibilité d'utiliser des modèles différents entre le contrôleur et le contrôlé, par exemple :
  - un modèle simplifié (*e.g.*, linéaire) du système pour le contrôleur, de manière à pouvoir dériver facilement une loi de commande, et
  - un modèle plus fidèle à la réalité pour le système contrôlé,
  - ce qui permettra de vérifier que la loi de commande dérivée du modèle simplifié sera efficace sur le système « réel ».

Deux niveaux de modélisation : test et contrôle

- Comment utiliser la modélisation hybride dans le développement des systèmes autonomiques ?
  1. Modéliser l'environnement et les éléments gérés de manière à les simuler pour les valider par une forme de test.
  2. Modéliser le contrôle pour calculer des politiques qui seront appliquées par les gestionnaires autonomiques, puis les simuler avec une politique donnée pour les valider.
- Plusieurs modèles hybrides peuvent devenir nécessaires :
  - *Un modèle plus complet et précis de l'entité adaptable dont fidélité est garante de la qualité de la validation du système.*
  - *Mais pour des raisons de calculabilité des politiques de contrôle, un modèle de contrôle peut être simplifié pour pour le résoudre par des techniques bien connues de la théorie du contrôle.*

Exemple : une entité adaptable peut être simulée comme un système hybride stochastique mais vue comme un système de contrôle linéaire déterministe.

Trajectoires observées versus modélisées



Par analogie avec les tests statistiques, deux types d'erreurs de modélisation (modèle/réel, modèle/modèle) :

- α : comportement licite non capturé par le modèle,
- β : comportement inexistant en réalité mais exhibé par le modèle.

Récapitulons...

1. L'auto-adaptabilité vue sous l'angle de l'informatique autonomique rend les entités logicielles auto-adaptables en leur adjoignant une entité qui réalise un ***contrôle en boucle fermée***.
2. La ***théorie du contrôle*** a développé des modèles permettant d'analyser et de trouver des lois de contrôle respectant les propriétés de ***stabilité***, de ***précision***, de ***temps de règlage court*** et de ***minimum de dépassement*** qui s'appliquent bien dans le contexte de l'informatique autonomique.
3. L'***automaticien*** se concentre le plus souvent sur des ***modèles continus déterministes*** et la ***stabilisation*** rapide d'un système pendant une action de contrôle.
4. Le spécialiste de la ***décision*** fait apparaître des coûts, cherche le plus souvent à capturer la nature ***stochastique*** des systèmes réels et applique des approches d'optimisation pour trouver des **lois de contrôle optimales**.
5. Le spécialiste des ***systèmes hybrides*** cherche à ***concilier*** les visions de l'automaticien et de l'informaticien, tout en appliquant des techniques de l'automaticien ou du spécialiste de la décision pour trouver les bonnes lois de contrôle.

Pour aller plus loin : sélection de lectures recommandées

- *Introduction to Control Theory and its Application to Computing Systems*, T. Abdelzaher, Y. Diao, J.L. Hellerstein, C. Lu, and X. Zhu, pp. 185–215.
- *Introduction to Discrete-Event Systems*, C.G. Cassandras et S. Lafortune, Springer, 2008, chapitres 3, 5.7 et 9.
- *A Unified Framework for Hybrid Control : Model and Optimal Control Theory*, M.S. Branicky, V.S. Borkar et S.K. Mitter, IEEE Transactions on Automatic Control, **43**, 1, Janvier 1998, pp. 31–45.