1

Introduction

1.1. Représentation d'état

Les systèmes mécaniques, biologiques, économiques, etc., qui nous entourent peuvent être souvent décrits par une équation différentielle du type :

$$\begin{cases} \dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) \\ \mathbf{y}(t) = \mathbf{g}(\mathbf{x}(t), \mathbf{u}(t)) \end{cases}$$

sous l'hypothèse que le temps t dans lequel évolue le système est continu [JAU 05]. Le vecteur $\mathbf{u}(t)$ est l'*entrée* (ou *commande*) du système. Sa valeur peut être choisie arbitrairement pour tout t. Le vecteur $\mathbf{y}(t)$ est la *sortie* du système et peut être mesuré avec une certaine précision. Le vecteur $\mathbf{x}(t)$ est appelé *état* du système. Il représente la mémoire du système, c'est-à-dire, l'ensemble des informations dont le système a besoin pour prédire son propre avenir, pour une entrée $\mathbf{u}(t)$ connue. La première des deux équations s'appelle *équation d'évolution*. Il s'agit d'une équation différentielle qui permet de savoir vers où va se diriger l'état $\mathbf{x}(t)$ sachant sa valeur à l'instant présent t et la commande $\mathbf{u}(t)$ que nous sommes en train d'exercer. La deuxième équation s'appelle *équation d'observation*. Elle permet de calculer le vecteur de sortie $\mathbf{y}(t)$, connaissant l'état et la commande à l'instant t. Attention, contrairement à l'équation d'évolution, cette équation n'est pas une équation différentielle car elle ne fait pas intervenir les dérivées des signaux. Les deux équations ci-dessus forment la *représentation d'état* du système.

Il est parfois utile de considérer un temps k discret, avec $k \in \mathbb{Z}$, où $\mathbb{Z}$ est l'ensemble des entiers relatifs. Si par exemple l'univers que nous considérons est un ordinateur, il est concevable de considérer que le temps k est discret et synchronisé

sur l'horloge du microprocesseur. Les systèmes à temps discret obéissent souvent à une équation de récurrence du type :

$$\begin{cases} \mathbf{x}(k+1) = \mathbf{f}(\mathbf{x}(k), \mathbf{u}(k)) \\ \mathbf{y}(k) = \mathbf{g}(\mathbf{x}(k), \mathbf{u}(k)) \end{cases}$$

Le premier objectif de ce livre est de comprendre le concept de représentation d'état à travers de nombreux exercices. Pour cela, nous allons considérer dans le chapitre 2 un grand nombre d'exemples variés et montrer comment arriver à une représentation d'état. Ensuite, nous montrerons au chapitre 3 comment simuler sur ordinateur un système donné par sa représentation d'état.

Le second objectif de ce livre est de proposer des méthodes pour *commander* les systèmes décrits par des équations d'état. C'est-à-dire que nous allons tenter de fabriquer des machines *automatiques* (où l'homme n'intervient quasiment pas, sauf pour donner ses ordres, ou *consignes*), appelées *régulateurs* capables de *domestiquer* (changer le comportement dans le sens que l'on souhaite) les systèmes considérés. Pour cela, le régulateur devra calculer les entrées $\mathbf{u}(t)$ à appliquer au système à partir de la connaissance (plus ou moins bruitée) des sorties $\mathbf{y}(t)$ et de la consigne $\mathbf{w}(t)$ (voir figure 1.1).

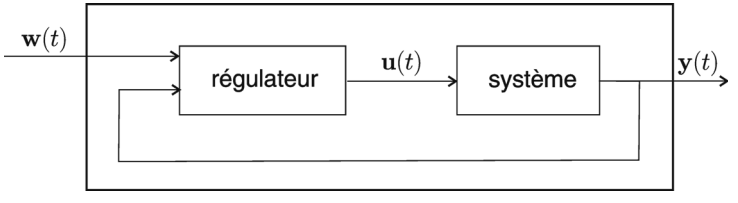


Figure 1.1. *Concept de boucle fermée illustrant la régulation d'un système*

Vu de l'utilisateur, le système, dit *système bouclé*, d'entrée $\mathbf{w}(t)$ et de sortie $\mathbf{y}(t)$ aura un comportement convenable. On dira que nous avons *asservi* le système. Dans cet objectif de régulation, nous allons, dans une première phase, nous intéresser uniquement aux systèmes linéaires, c'est-à-dire que les fonctions $\mathbf{f}$ et $\mathbf{g}$ sont supposées linéaires. Ainsi, dans le cas du temps continu, les équations d'état du système s'écrivent :

$$\begin{cases} \dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \\ \mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t) \end{cases}$$

et dans le cas du temps discret, elles deviennent :

$$\begin{cases} \mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) \\ \mathbf{y}(k) = \mathbf{C}\mathbf{x}(k) + \mathbf{D}\mathbf{u}(k) \end{cases}$$

Les matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}$ sont appelées *matrices d'évolution*, *de commande*, *d'observation* et *directe*. Une analyse détaillée de ces systèmes sera faite au chapitre 4. Nous expliquerons ensuite, dans le chapitre 5 comment stabiliser ces systèmes. Enfin, nous montrerons, au chapitre 6, qu'autour de certains points, dits *de fonctionnement*, les systèmes non linéaires se comportent comme des systèmes linéaires. Il sera alors possible de les stabiliser par les même méthodes que celles développées pour le cas linéaire.

Enfin, ce livre est accompagné de nombreux programmes MATLAB disponibles à l'adresse suivante :

http://www.ensta-bretagne.fr/jaulin/isteauto.html

1.2. Exercices

Exercice 1. Robot sous-marin

Le robot sous-marin *Saucisse* de l'ENSTA Bretagne [JAU 09], dont la photo est donnée sur la figure 1.2, est un système asservi. Il comporte un ordinateur, trois propulseurs, une caméra, une boussole et un sonar. A quoi correspond dans ce contexte le vecteur d'entrée $\mathbf{u}$, le vecteur de sortie $\mathbf{y}$, le vecteur d'état $\mathbf{x}$ et la consigne $\mathbf{w}$? Où intervient l'ordinateur dans la boucle de commande ?

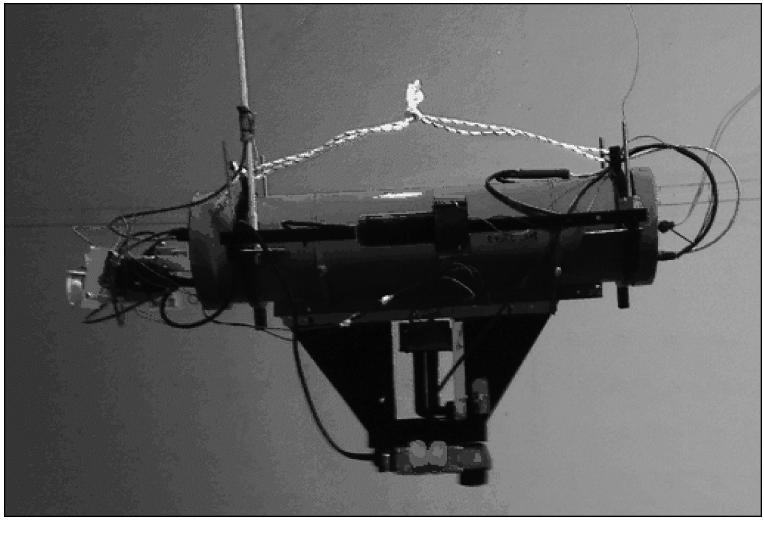
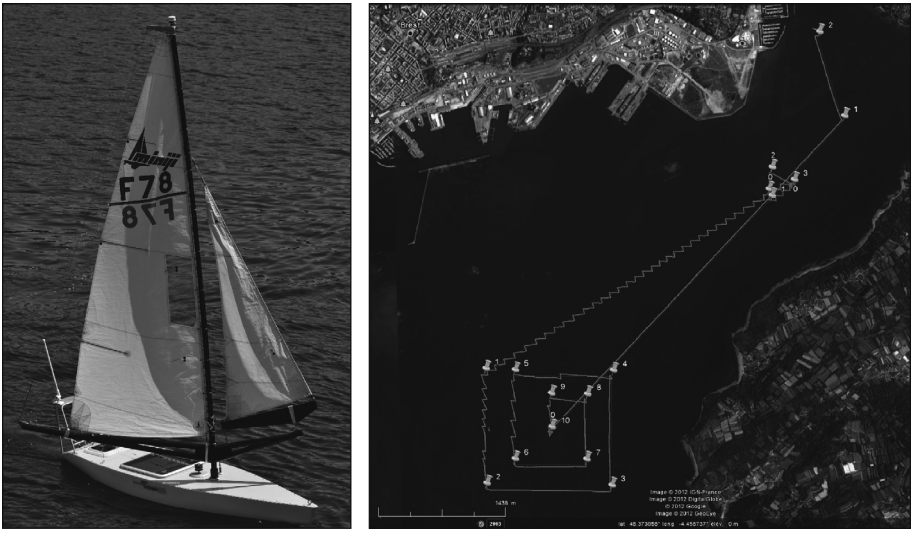


Figure 1.2. *Robot sous-marin asservi*

Exercice 2. Robot voilier

Le robot voilier *Vaimos* (IFREMER et ENSTA Bretagne) de la figure 1.3 est aussi un système asservi [GOR 11, JAU 12a, JAU 12b]. Il est capable d'accomplir seul des trajectoires du type de celle dessinée sur la figure 1.3. Il dispose d'un gouvernail et d'une voile réglable par une écoute. Il possède un anémomètre en sommet du mât, une boussole et un GPS. Décrire à quoi peut correspondre le vecteur d'entrée $\mathbf{u}$, le vecteur de sortie $\mathbf{y}$, le vecteur d'état $\mathbf{x}$ et la consigne $\mathbf{w}$.



a) b)

Figure 1.3. *Robot voilier Vaimos a) et une trajectoire faite par Vaimos b). Les zig-zags de la trajectoire réalisée viennent du fait que Vaimos a dû louvoyer pour remonter le vent*

1.3. Corrections

Correction de l'exercice 1 (robot sous-marin)

Le vecteur d'entrée $\mathbf{u} \in \mathbb{R}^3$ correspond à la tension électrique donnée aux trois propulseurs et le vecteur de sortie $\mathbf{y}(t)$ comprend la boussole, les données sonars et les images prises par les caméras. Le vecteur d'état $\mathbf{x}$ correspond à la position, à l'orientation et aux vitesses du robot. La consigne $\mathbf{w}$ est demandée par le superviseur. Par exemple si l'on veut faire une commande en cap, la consigne $\mathbf{w}$ sera la vitesse et le cap désiré pour le robot. Le régulateur se trouve être un programme exécuté par l'ordinateur.

Correction de l'exercice 2 (robot voilier)

Le vecteur d'entrée $\mathbf{u} \in \mathbb{R}^2$ correspond à la longueur de l'écoute $\delta_v^{\max}$ et à l'angle du gouvernail δ_g. Le vecteur de sortie $\mathbf{y} \in \mathbb{R}^4$ comprend les données GPS $\mathbf{m}$, les données de l'anémomètre à ultrasons (girouette en haut du mât) ψ et la boussole θ. La consigne $\mathbf{w}$ indique ici le segment $\mathbf{ab}$ à suivre. La figure 1.4 illustre cette boucle de régulation. Un superviseur, non représenté sur la figure, se charge de séquencer les segments à suivre de façon à ce que le robot réalise la trajectoire désirée (ici 12 segments formant une spirale carrée suivie d'un retour au port).

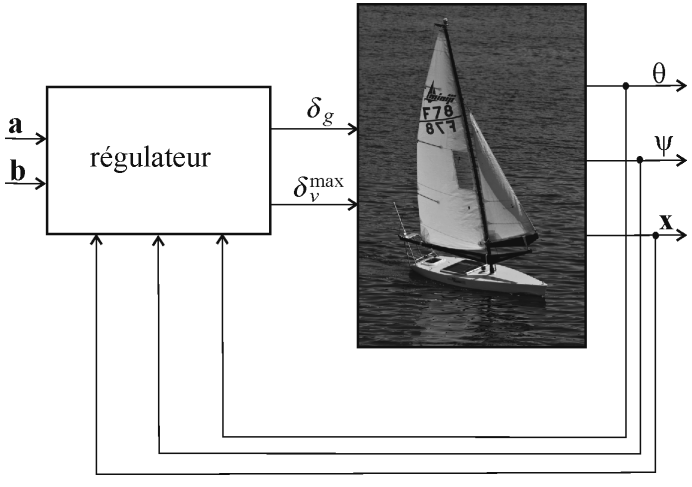


Figure 1.4. *Boucle de régulation du robot voilier*