Avant-propos

Cet ouvrage étudie en profondeur les techniques avancées de conception ainsi que la technologie moderne de mise en œuvre des systèmes de commande et régulation assistée par ordinateur de processus dynamiques déterministes et stochastiques.

Il s'adresse aux acteurs (étudiants, enseignants et chercheurs) d'écoles d'ingénieurs, d'écoles de formation de formateurs à l'enseignement technique, d'écoles doctorales et de centres de recherche en sciences appliquées.

En effet, les lecteurs y découvriront :

– des techniques de construction des modèles d'état discrets canoniques de processus dynamiques, ainsi que les méthodes de conception de régulateurs numériques par retour d'état discret ;

– une étude de cas détaillée du projet de création et de mise en œuvre effective d'un nouveau banc d'essai multimédia assisté par ordinateur de servomécanismes, à base de palettes virtuelles de contrôleurs PIDF (proportionnel, intégral et différentiel avec filtre), de régulateurs par retour d'état avec ou sans observateur et d'instruments virtuels ;

– des schémas algorithmiques détaillés de systèmes de commande optimale déterministe ou stochastique, avec temps d'optimisation fini ou infini ;

– des secrets de création et de prototypage d'une nouvelle plateforme virtuelle Matlab/GUI déportée, de conception rapide de systèmes de commande optimale déterministe et stochastique ;

– des topologies infrastructurelles de systèmes de commande et régulation télé-opérables en temps réel ;

– une étude de cas détaillée de création et de mise en œuvre effective d'un nouveau laboratoire d'automatique télé-opérable (LABAUTOP) en temps réel *via* Internet ;

– des programmes Matlab didactiques, permettant de reproduire en cas de besoin les résultats numériques et graphiques présentés dans l'ouvrage ;

– des exercices corrigés situés à la fin de chaque chapitre, en vue de consolider les connaissances techniques acquises.

Le contenu de cet ouvrage résulte de l'expérience acquise par l'auteur au cours des quinze dernières années à l'ENSET (École normale supérieure d'enseignement technique) et à l'UFD (Unité de formation doctorale) « sciences de l'ingénieur » de l'université de Douala, dans le cadre de ses activités combinées :

– d'enseignement des cours magistraux de « commande optimale déterministe et stochastique » et de « programmation avancée assistée par Matlab » ;

– de recherche scientifique de nouvelles plateformes didactiques flexibles ;

– d'appui au développement de la technologie de réglage assisté par ordinateur en ingénierie moderne de procédés automatiques.

L'auteur tient à louer la prime de recherche scientifique de l'État camerounais *via* le ministère de l'Enseignement supérieur, qui lui a permis de financer une partie des coûts de préparation et de rédaction de cet ouvrage.

L'auteur remercie sincèrement :

– Pr Womonou Robert, directeur et promoteur d'ESSET de Douala et de Nkongsamba, qui l'a motivé à finaliser cet ouvrage ;

– Pr Nneme Nneme Léandre, directeur et chef du département « génie informatique » de l'ENSET de Douala, qui a participé aux travaux d'étude du laboratoire d'automatique télé-opérable, présenté dans le chapitre 8 ;

– Dr Pauné Félix, chargé de cours au département « génie informatique » de l'ENSET de Douala, qui est l'auteur principal et l'administrateur système dudit laboratoire d'automatique télé-opérable, étudié et mis en œuvre dans le cadre de sa thèse de doctorat/PhD, dirigée par l'auteur de cet ouvrage ;

– Dr Lonla, chargé de cours au département « génie informatique » de l'ENSET de Douala, qui a été le premier télé-opérateur d'essai sans assistance en ligne dudit laboratoire d'automatique télé-opérable à partir d'un pays étranger, lors de son séjour post-doctoral à Lens en France ;

– les membres de l'équipe éditoriale de ISTE Éditions, pour leur excellente collaboration dans toutes les phases d'édition de cet ouvrage ;

– son épouse, Mme Mbihi, née Tsafack Pélagie Marthe, qui a déployé à son endroit une assistance rapprochée, ainsi que tous ceux qui ont beaucoup contribué aux travaux de production de cet ouvrage.

Introduction

I.1. Architecture des systèmes de réglage assisté par ordinateur

L'architecture générale d'un système complet de réglage assisté par ordinateur, est représentée à la figure I.1, où les principaux sous-ensembles constitutifs observés sont désignés comme suit :

– processus dynamique réel à régler ;

– interface d'acquisition de données multifonctions (IADM) ;

– ordinateur PC/multimédia de commande numérique rapprochée ;

– terminaux de réglage à distance du processus réel *via* Internet.

Ces sous-ensembles constitutifs sont étudiés en détail dans les prochaines sections.

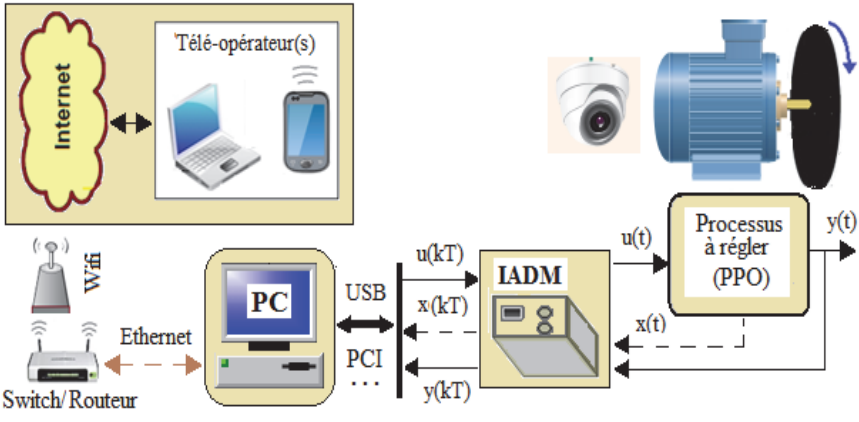


Figure I.1. *Architecture d'un système complet de réglage assisté par ordinateur*

I.2. Processus dynamiques à régler

Le processus dynamique réel à régler correspond à la partie puissance et opérative (PPO) d'un système asservi en boucle ouverte. Dans la PPO, les notations u, x et y désignent respectivement les grandeurs physiques de commande rapprochée, d'état et de sortie. Ces grandeurs sont naturellement des variables en temps continu.

I.3. Interface d'acquisition de données multifonctions (IADM)

Une IADM est en réalité un macrocontrôleur (ou système unifié à microcontrôleur). Elle joue le rôle d'interpréteur de protocole de communication entre le processus dynamique analogique et un ordinateur à vocation numérique.

L'étude détaillée des IADM modernes constitue un vaste sujet d'actualité en informatique industrielle (Mbihi et Motto, 2012). Nous nous limiterons ici au résumé des éléments de connaissance stratégiques permettant de maîtriser les critères de choix ainsi que le schéma opérationnel de programmation en temps réel d'une IADM en automatique et informatique industrielle.

I.3.1. *Microbus d'entrée/sortie*

Une IADM utilisée en technologie de commande et régulation assistée par ordinateur dispose d'un système spécifique de microbus d'entrée/sortie. Le tableau I.1 décrit les types de microbus utilisés en instrumentation assistée par ordinateur.

Classe	Type	Année	Paquet (*) D, C	Débit maximal	Portée
Ports	RS232	1962	8, 3	7 Ko/s	30 m
	LPT	1992	8, 0	2 Mo/s	3 m
	USB	1995	1024, 1027	1,5 Go/s	1,8 m
	Ethernet	1980	-	40 Gb/s	100 m
Slots	PCI	1992	32, 0	132 Mo/s	Faible
	PCI-X	1999	32, 0	160 Mo/s	Faible
			64, 0	320 Mo/s	Faible
	PCI-E	2004	128, 130	8 Go/s	Faible
Autres	Wifi	1997	-	11 Mbits/s	400 m

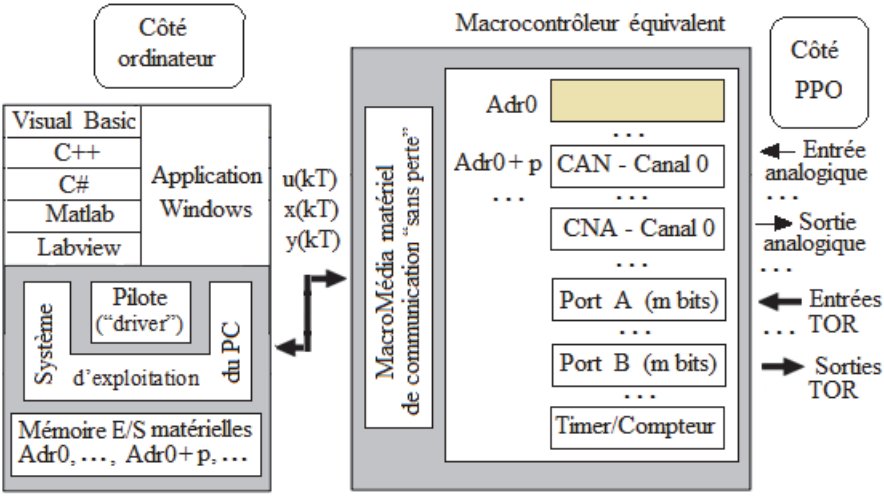
(*) D (bits) : données par paquet ; C (bit) : contrôle par paquet

Tableau I.1. *Microbus utilisés en instrumentation assistée par ordinateur*

En pratique, les microbus RS232 et LPT ne sont plus d'actualité. En revanche, les cartes mères d'ordinateurs PC/multimédias supportent les microbus PCI, PCI-X, PCI-E et Wifi.

I.3.2. *Structure logicielle unifiée*

En micro-informatique, la structure logicielle d'un microsystème désigne un schéma descriptif simplifié des constituants programmables internes qui sont accessibles en temps réel par un programme d'application. Ainsi, la figure I.2 présente la structure logicielle d'une IADM vue du côté ordinateur et du côté PPO (partie puissance et opérative) du processus dynamique.



PPO : partie puissance et opérative ; TOR : tout ou rien ; CAN : convertisseur analogique/numérique ; CNA : convertisseur numérique/analogique

Figure I.2. *Structure logicielle unifiée d'une IADM*

Autrement dit, la chaîne matérielle constituée de la carte mère de l'ordinateur, suivie du connecteur et du contrôleur de bus de l'IADM, est modélisable par un macromédia de communication sans perte entre l'environnement logiciel de l'ordinateur et les ressources d'instrumentation programmables de l'IADM.

Les principaux types de ressources d'instrumentation programmables, apparentes dans la structure logicielle des IADM modernes, sont les suivants :

– convertisseur analogique/numérique (CAN) ;

– convertisseur numérique/analogique (CNA) ;

– module d'entrée spécialisé (pour thermocouple, encodeurs optiques, etc.) ;

– module de sortie spécialisé MLI (modulation en largeur d'impulsions) ou PWM (*Pulse Width Modulation*) en anglais ;

– temporisateur de gestion de tâches périodiques (*timer* en anglais) ;

– ports d'entrée/sortie TOR (tout ou rien), Bluetooth, Wifi, etc.

Chaque ressource programmable d'une IADM est accessible en temps réel du côté logiciel d'application Windows par adressage relatif *via* une zone mémoire de l'ordinateur allouée aux cibles d'entrée/sortie matérielle. Cette zone débute à une adresse de base d'installation « Ard0 » affectée automatiquement après détection de l'IADM par le système d'exploitation.

Ainsi, les grandeurs caractéristiques de commande $u(t)$, d'état $x(t)$ et de sortie $y(t)$ d'un processus dynamique sont exprimées du côté ordinateur sous forme de quantités discrètes respectives $u(k\mathrm{T})$, $x(k\mathrm{T})$ et $y(k\mathrm{T})$, T étant la période d'échantillonnage.

I.3.3. *Schéma opérationnel de programmation en temps réel*

Le schéma opérationnel historique de programmation en temps réel d'une IADM est présenté dans la figure I.3, où sont illustrées les possibilités de compatibilité entre les générations d'outils de développement logiciel et les systèmes d'exploitation d'ordinateur.

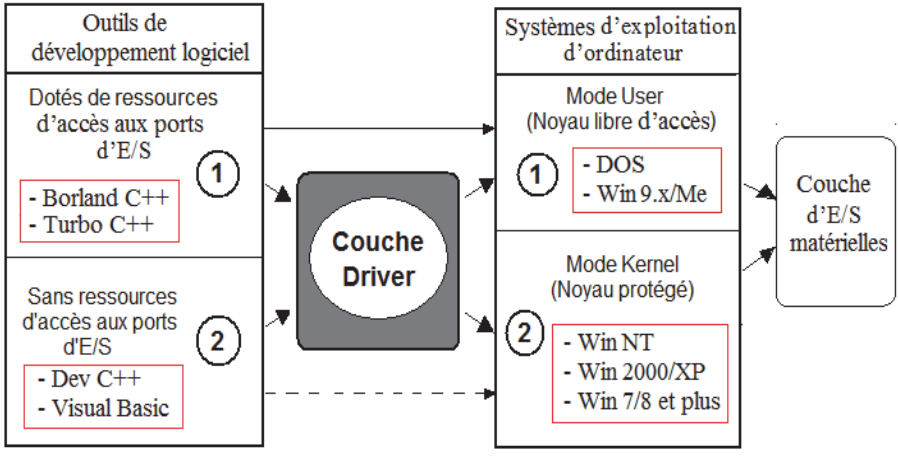


Figure I.3. *Schéma opérationnel de programmation en temps réel d'une IADM*

En effet, l'histoire de la programmation en temps réel des IADM révèle :

– deux générations d'outils de développement logiciel, qui se distinguent par la disponibilité des ressources d'accès aux ports d'entrée/sortie matérielle. Ainsi, Turbo C++ et Borland C++ sont des exemples d'outils de développement historiques dotés de fonctions d'adressage de ports d'entrée/sortie matérielle, ce qui n'est pas le cas de Dev C++ et Visual Basic ;

– deux générations de systèmes d'exploitation, qui se distinguent par les conditions de sécurité d'accès effectif au noyau (*kernel* en anglais) d'adressage de ports d'entrée/sortie matérielle. Ainsi, DOS et Windows 9.x/Me sont des exemples de système d'exploitation historiques dotés d'un noyau dont les conditions d'accessibilité ne sont pas sécurisées, ce qui n'est pas le cas pour Windows NT, Windows 2000, Windows XP, etc.

Ces constats historiques permettent donc d'envisager les quatre cas d'association outils de développement/systèmes d'exploitation suivants :

– Cas 1 → 1 : techniquement intéressant, mais qui n'est plus d'actualité ;

– Cas 1 → 2 : problématique du point vue de l'adressage direct de ports ;

– Cas 2 → 1 : plus problématique que le précédent ;

– Cas 2 → 2 : irréaliste.

En pratique, la solution aux problèmes d'adressage en temps réel des ports matériels consiste à utiliser un pilote logiciel (*driver* en anglais) livré par le fabricant de l'IADM à utiliser.

I.3.4. *Pilote logiciel d'une IADM*

En programmation Windows, le pilote logiciel d'une IADM est une catégorie de bibliothèques de liens dynamiques (*.DLL) qui mettent à disposition des palettes de fonctions spécialisées de gestion des ressources d'instrumentation offertes.

Les fonctions d'instrumentation d'une IADM sont exécutables en temps réel en mode « Kernel » sur appel import/export à partir d'un programme d'application dûment développé avec tout outil de programmation recommandé dans le manuel d'utilisation du produit.

Le tableau I.2 décrit des exemples de fonctions d'instrumentation intégrées dans le pilote K8055.dll de la carte USB/VM110 de Velleman. Ce pilote K8055.dll est destiné aux outils populaires de programmation Windows en langage Delphi, Visual Basic, C++ et C#.

Description		Syntaxe de déclaration dans un programme Visual Basic	Exemple d'utilisation dans un programme Visual Basic
Rôle	**Noms dans K8055.dll**		
Ouvrir	*OpenDevice*	Private Declare Function *OpenDevice* Lib "k8055d.dll" (ByVal CardAddress As Long) As Long	Si carte 0 installée : *OpenDevice* (0)
Fermer	*CloseDevice*	Private Declare Sub *CloseDevice* Lib "k8055d.dll" ( )	*CloseDevice* ( )
Conversion A/N	*ReadAnalogChannel*	Private Declare Function *ReadAnalogChannel* Lib "k8055d.dll" (ByVal Channel As Long) As Long	*Ncan* = *ReadAnalogChannel*(1)
Conversion N/A	*OutputAnalogChannel*	Private Declare Sub *OutputAnalogChannel* Lib "k8055d.dll" (ByVal Channel As Long, ByVal Data As Long)	*'Ncna* = 254 : *'N° du canal = 1* *OutputAnalogChannel* (1, *Ncna*)

A/N : analogique/numérique ; N/A : numérique/analogique

Tableau I.2. *Exemples de fonctions de base du pilote d'une IADM : cas du pilote K8055.dll de la carte USB/VM110*

I.3.5. *Conversions A/N et N/A dans un programme d'instrumentation*

Dans le tableau I.2, il est important de constater qu'en général la fonction de conversion A/N (analogique/numérique) du pilote de l'IADM considérée retourne en réalité au programme appelant d'instrumentation la valeur entière *Ncan* du code d'une tension *y* à mesurer. La mesure effective devra donc être reconstituée numériquement dans le programme d'instrumentation sur la base de la caractéristique d'entrée/sortie du CAN (convertisseur A/N) réel opérant dans l'IADM.

Ainsi, dans le cas d'un CAN opérant dans une plage de tension $[U_m\ U_M]$ par codage binaire décalé de *n* bits, la mesure effective *y* représentée par un code entier *Ncan* peut être reconstituée dans le programme appelant d'instrumentation par la formule :

$$y = \left(\frac{U_M - U_m}{2^n} \right) \text{Ncan} + U_m \qquad [I.1]$$

Par ailleurs, si la mesure de y, donnée par [I.1], correspond en réalité à la tension de sortie du capteur d'une grandeur physique quelconque v, alors il suffirait de reconstituer la valeur effective de v en utilisant la caractéristique inverse $v(y)$ dudit capteur.

De façon analogue, l'argument de la fonction CNA d'un pilote correspond en général à la valeur entière *Ncna* d'un code numérique de n bits à appliquer à l'entrée du CNA opérant dans l'IADM. Dans les conditions opératoires théoriquement réciproques à [I.1], cette valeur codée *Ncna* devra donc être reconstituée à partir de la tension réelle u de commande numérique par la formule :

$$N_{cna} = E\left(\frac{u - U_m}{q}\right) \qquad [I.2]$$

avec $q = \frac{U_M - U_m}{2^n}$ et où E(.) désigne l'opérateur qui extrait la partie entière à gauche de son argument. La grandeur q définie dans [I.2] désigne le *quantum* du convertisseur et correspond à la plus petite valeur qui peut être convertie avec précision.

Pour les convertisseurs unipolaires, on a généralement $U_m = 0$ et $U_M < 0$ dans [I.1] et [I.2]. En revanche, les convertisseurs bipolaires les plus populaires offrent une plage opératoire symétrique, ce qui implique $U_M = -U_m = U > 0$ dans [I.1] et [I.2].

Les technologies de mise en œuvre des conversions A/N et N/A sont présentées en détail dans les travaux de Mbihi (2012) et Moffo et Mbihi (2015).

I.3.6. *Autres informations pratiques sur les IADM*

Les informations pratiques complémentaires sur les pilotes logiciels pour IADM se résument comme suit :

– le pilote logiciel d'une IADM peut être développé par divers types d'outils de programmation tels que Visual C++, Visual C#, Builder C++, Matlab, Labview, etc. Cependant, ce développement nécessite des compétences avancées en programmation Windows ;

– en pratique, chaque IADM est livrée avec son pilote logiciel disponible pour téléchargement gratuit dans le site Web du fabricant. Par ailleurs, les fonctions d'instrumentation d'un pilote logiciel sont exploitables par tout programme externe développé avec un outil de programmation recommandé dans le manuel d'utilisation dudit pilote ;

– le fabricant d'une IADM fournit aussi généralement des exemples de programmes, de test des fonctions d'instrumentation du pilote logiciel de son produit. Ces programmes

à vocation didactique constituent le point de départ de développement des applications avancées de commande et régulation assistée par ordinateur des processus dynamiques ;

– le pilote logiciel C++ de base d'une IADM peut être utilisé comme brique logicielle pour la construction d'un nouveau pilote étendu, destiné à un outil de programmation de plus haut niveau. On verra au chapitre 3 les secrets de création d'un pilote MEX-C++ pour applications Matlab à partir du pilote de base C++ de la carte USB/VM110 de Velleman.

I.4. Ordinateur PC/multimédia

L'ordinateur PC/multimédia est une machine informatique équipée de ressources matérielles et logicielles suffisantes pour la production et l'exploitation en temps réel des applications multimédias.

Dans un contexte de commande et régulation assistée par calculateur numérique, un ordinateur PC/multimédia a pour tâches principales de :

– collecter et reconstituer les grandeurs d'entrée issues de l'interface homme/machine (IHM) et de l'IADM ;

– calculer et formater les valeurs de commande numérique à appliquer aux ports de sortie matérielle ;

– mettre à jour les constituants de sortie de l'IHM (tableurs de données, boîtes de message, composants graphiques, composants audio/vidéo, indicateurs d'états, etc.) ;

– gérer les bases de données ainsi que le journal de bord de la production ;

– gérer les services Web en cas de nécessité.

I.5. Terminaux d'accès à distance

Les terminaux observés sur la figure I.1 sont des équipements informatiques câblés ou sans fil permettant aux télé-opérateurs autorisés de régler à distance un processus dynamique réel *via* un réseau informatique local ou Internet.

On verra dans les chapitres 7 et 8 qu'en instrumentation virtuelle l'équipement logiciel minimum d'un terminal de réglage à distance de processus dynamiques *via* Internet est un navigateur Web (Internet Explorer, Mozilla, etc.). Cependant, selon la technologie et l'outil de mise en œuvre du système serveur tournant dans l'ordinateur local, un moteur d'exécution compatible pourrait aussi être installé du côté terminal.

I.6. Organisation de l'ouvrage

Cet ouvrage est organisé en trois parties, chaque partie étant structurée en chapitres cohérents.

La première partie présente les éléments avancés ainsi qu'un banc d'essai multimédia de commande et régulation assistée par ordinateur de servomécanismes. Ainsi, les modèles d'état discrets canoniques de processus dynamiques sont étudiés dans le chapitre 1, puis la conception des systèmes de commande et régulation numérique par retour d'état est présentée en détail dans le chapitre 2. Par ailleurs, les secrets de construction d'un banc d'essai multimédia de commande et régulation assistée par ordinateur de servomécanismes sont dévoilés dans le chapitre 3.

La deuxième partie porte sur la commande et régulation numérique optimale déterministe et stochastique. Le chapitre 4 de cette partie permettra aux lecteurs d'acquérir les éléments fondamentaux, ainsi que les principaux résultats de la théorie de commande et régulation numérique optimale des processus dynamiques déterministes. Ils trouveront ensuite dans le chapitre 5 une revue claire de la théorie de commande et régulation optimale de processus dynamiques stochastiques. Ensuite, ils découvriront dans le chapitre 6 une méthodologie d'étude du projet de création et de mise en œuvre effective d'une nouvelle plateforme Matlab/GUI déployée de conception et simulation virtuelle assistée par ordinateur pour des systèmes de commande optimale stochastique, sur horizon de temps fini ou infini.

La troisième partie porte sur les systèmes de commande et régulation assistée par ordinateur télé-opérables *via* Internet. Dans le chapitre 7 de cette dernière partie, le lecteur se familiarisera aux topologiques infrastructurelles des systèmes de régulation à distance. Il découvrira ensuite au chapitre 8 les secrets de conception et de création effective d'un nouveau laboratoire d'automatique télé-opérable (LABAUTOP) *via* Internet.

Enfin, il est aussi important de rappeler que les exercices corrigés sont situés à la fin de chaque chapitre de l'ouvrage. Par ailleurs, les lecteurs trouveront en annexe 1 une table de transformées en z, et, en annexe 2, un tableau récapitulatif des éléments Matlab utilisés dans les programmes didactiques présentés dans l'ouvrage. Enfin, l'annexe 3 met à leur disposition un résumé des résultats de méthodes de discrétisation des fonctions de transfert de processus dynamiques et régulateurs analogiques PIDF (proportionnel, intégral, différentiel avec filtre).