

Titre: Conception et développement d'un système d'acquisition et de
Title: contrôle à distance pour la fabrication de matériaux composites

Auteur: Chakib Boushel
Author:

Date: 2009

Type: Mémoire ou thèse / Dissertation or Thesis

Référence: Boushel, C. (2009). Conception et développement d'un système d'acquisition et
Citation: de contrôle à distance pour la fabrication de matériaux composites [Master's
thesis, École Polytechnique de Montréal]. PolyPublie.
<https://publications.polymtl.ca/8298/>

 **Document en libre accès dans PolyPublie**
Open Access document in PolyPublie

URL de PolyPublie: <https://publications.polymtl.ca/8298/>
PolyPublie URL:

**Directeurs de
recherche:** Edu Ruiz
Advisors:

Programme: Unspecified
Program:

UNIVERSITÉ DE MONTRÉAL

CONCEPTION ET DÉVELOPPEMENT D'UN SYSTÈME D'ACQUISITION ET DE
CONTRÔLE À DISTANCE POUR LA FABRICATION DE MATÉRIAUX
COMPOSITES

CHAKIB BOUSHEL

DÉPARTEMENT DE GÉNIE MÉCANIQUE

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION
DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES
(GÉNIE MÉCANIQUE)

Mars 2009

© Boushel Chakib, 2009.



Library and
Archives Canada

Bibliothèque et
Archives Canada

Published Heritage
Branch

Direction du
Patrimoine de l'édition

395 Wellington Street
Ottawa ON K1A 0N4
Canada

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file *Votre référence*
ISBN: 978-0-494-49431-8
Our file *Notre référence*
ISBN: 978-0-494-49431-8

NOTICE:

The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

AVIS:

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protègent cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

■*■
Canada

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé :

CONCEPTION ET DÉVELOPPEMENT D'UN SYSTÈME D'ACQUISITION ET DE
CONTRÔLE À DISTANCE POUR LA FABRICATION DE MATÉRIAUX
COMPOSITES

présenté par : BOUSHEL Chakib

en vue de l'obtention du diplôme de Maîtrise ès sciences appliquées

a été dûment accepté par le jury d'examen constitué de :

M. BALAZINSKI, Marek, Ph.D., président

M. RUIZ Eduardo Antonio Julian, Ph.D., membre et directeur de recherche

M. TROCHU François, Ph.D., membre

À ma famille

REMERCIEMENTS

Je tiens à remercier mon directeur de recherche M. Edu Ruiz pour sa supervision effective tout au long du déroulement de ma maîtrise. La liberté et la marge de manœuvre dont j'ai bénéficié ont été une preuve incontestable de la confiance continuelle dont il m'a fait part.

Je remercie aussi M. François Trochu, titulaire la Chaire des Composites à Haute Performance (CCHP), pour m'avoir permis de découvrir le domaine des composites et d'y avoir mis en pratique mes connaissances et mes compétences.

Je remercie aussi le président du jury, M Marek Balazinsky pour avoir accepté d'évaluer mon mémoire de maîtrise.

Je salue M. Vincent Achim, étudiant doctorant au sein de la chaire CCHP, pour son assistance pratique et sa disponibilité.

Mes hommages vont à l'ensemble des étudiants de la chaire CCHP, que j'ai eu tant de plaisir à connaître et à apprécier.

RÉSUMÉ

Les systèmes d'acquisition de données et de contrôle font partie intégrante des ressources matérielles industrielles actuelles. L'industrie des matériaux composites est un exemple concret de domaines ayant recours de façon intensive à ce type de systèmes. Le caractère critique qui qualifie certaines branches de cette industrie impose des critères stricts quant à la qualité des procédés et des produits qui s'y rattachent. La conformité à de telles contraintes reste étroitement liée à la fiabilité des ressources matérielles utilisées en l'occurrence les outils d'acquisition et de contrôle. Par ailleurs, le caractère multidisciplinaire de l'industrie des composites incite à introduire des moyens de collaboration efficace entre les divers acteurs potentiels du domaine. La réalisation d'une telle avancée reste possible au travers de l'exploitation de l'infrastructure des réseaux informatiques et des technologies qui y sont relatives.

Le présent projet a pour objectif de proposer une solution d'acquisition de données et de contrôle thermique à distance. Cette solution s'applique à la fabrication des matériaux composites en général, et à la caractérisation thermique des résines thermodurcissables en particulier.

ABSTRACT

Data acquisition and control systems remain inherent parts of industry-related resources. Composites materials manufacturing represents a relevant field within which data acquisition and control tools are intensively exploited. Some composites-related industrial sectors require stringent standards to be applied upon related processes and products. Compliance with such standards remains closely dependent upon utilized-resources reliability and accuracy features.

Being a multi-disciplinary manufacturing environment, composites industry needs enhanced and optimized collaboration tools to be built up in order to improve related-processes efficiency. That said, reliance upon networking infrastructure and its related technologies seems to be an interesting choice to be taken into account within data acquisition and control systems development.

The current work provides a new solution for remote data acquisition and process control. This solution has been applied on composites manufacturing in general and thermosetting resins thermal characterization in particular.

TABLE DES MATIÈRES

DÉDICACE.....	IV
REMERCIEMENTS	V
RÉSUMÉ.....	VI
ABSTRACT	VII
TABLE DES MATIÈRES	VIII
LISTE DES FIGURES	XII
LISTE DES ANNEXES	XVI
NOTATIONS	XVII
INTRODUCTION	1
CHAPITRE 1 : REVUE BIBLIOGRAPHIQUE	3
1.1. Systèmes répartis sur réseaux informatiques	3
1.1.1. Avantages de systèmes répartis	3
1.1.2. Architectures de systèmes répartis	4
1.1.3. Technologies de répartition des systèmes	8
1.2. Systèmes d'acquisition de données	14
1.2.1. Éléments matériels de systèmes d'acquisition	16
1.2.2. Logiciels d'acquisition de données	17
1.3. Commande de procédés par ordinateur	18
1.4. Tendances des systèmes d'acquisition de données et de contrôle	21

CHAPITRE 2 : SYSTÈME MATÉRIEL D'ACQUISITION ET DE CONTRÔLE THERMIQUES.....	24
2.1. Architecture du système d'acquisition et de contrôle thermique	24
2.1.1. Capteurs	26
2.1.2. Actionneurs	29
2.1.3. Microcontrôleur.....	33
2.2. Communication matériel-logiciel.....	35
2.2.1. Requête PC à DAC	36
2.2.2. Réponse DAC à PC	38
2.2.3. Synchronisation de la communication.....	38
CHAPITRE 3 : ANALYSE LOGICIELLE DU SYSTÈME D'ACQUISITION ET DE CONTRÔLE.....	40
3.1. Requis fonctionnels.....	40
3.1.1. Spécifications générales.....	40
3.1.2. Requis d'acquisition de données et de contrôle	42
3.1.3. Enregistrement des données et des configurations	44
3.2. Requis qualitatifs.....	46
3.2.1. Robustesse.....	46
3.2.2. Interactivité	47
3.2.3. Évolutivité et maintenabilité	48

CHAPITRE 4 : CONCEPTION LOGICIELLE DU SYSTÈME D'ACQUISITION ET DE CONTRÔLE.....	49
4.1. Architecture globale	49
4.2. Processus de localisation	53
4.3. Interface utilisateur.....	55
4.3.1. Enregistrement de l'utilisateur	55
4.3.2. Configuration des modules matériel.....	57
4.3.3. Configuration du contrôle du procédé	58
4.3.4. Configuration des consignes de contrôle	59
4.3.5. Fenêtre principale	61
4.4. Architecture orientée services.....	63
4.5. Environnement et plateforme	65
4.6. Détails de conception logicielle.....	66
4.6.1. Application Serveur	67
4.6.2. Application cliente.....	75
CHAPITRE 5 : ÉTUDE EXPÉRIMENTALE ET APPLICATIONS.....	79
5.1. Stratégie de contrôle thermique.....	79
5.1.1. Configuration de contrôle	79
5.1.2. Optimisation des paramètres de contrôle	82
5.2. Application du système DAC pour la caractérisation industrielle d'une résine thermodurcissable.....	86

5.2.1.	Configuration de l'application	86
5.2.2.	Résultats	88
5.2.3.	Conclusion	93
CONCLUSIONS ET PERSPECTIVES.....		94
RÉFÉRENCES		97
ANNEXES.....		100

LISTE DES FIGURES

Figure 1.1 Exemple d'architecture pair à pair.....	5
Figure 1.2 Exemple d'architecture client/serveur.....	6
Figure 1.3 Architecture hybride d'un système de partage de fichiers	8
Figure 1.4 Exemple d'architecture de système réparti fondé sur un intergiciel orienté-objets [7].....	11
Figure 1.5 Diagramme d'un système d'acquisition [16].....	15
Figure 1.6 Boucle typique de commande de procédé par ordinateur	19
Figure 1.7 Critères d'évaluation du contrôle des systèmes	21
Figure 2.1 Vue externe du système matériel DAC	25
Figure 2.2 Vue interne du système matériel DAC.....	25
Figure 2.3 Architecture matérielle globale du système DAC	26
Figure 2.4 Schéma de la structure typique d'un thermocouple	27
Figure 2.6 Photographie des capteurs thermocouple et thermoflux	28
Figure 2.5 Schéma de la structure d'un capteur de flux thermique [24].....	28
Figure 2.7 Photographie de l'élément chauffant	30
Figure 2.8 Photographie de la valve d'air comprimé.....	30
Figure 2.9 Évolution du débit d'air induit à la sortie de la valve en fonction du courant électrique imposé pour une pression d'air à l'entrée de l'ordre de 100 psi.	31

Figure 2.10 Linéarisation du rapport du débit d'air induit à la sortie de la valve et du courant électrique imposé pour une pression d'entrée de l'air de l'ordre de 100 psi.	32
Figure 2.11 Structure du protocole de communication.....	36
Figure 2.12 Synchronisation de la communication entre le PC et la partie matérielle du système DAC.....	39
Figure 3.1 Diagramme des fonctionnalités globales du système logiciel d'acquisition et de contrôle	41
Figure 3.2 Diagramme de séquence relatif aux cas d'utilisation de l'acquisition de données et du contrôle.....	43
Figure 3.3 Diagramme de séquence pour l'exportation de données.....	45
Figure 3.4 Diagramme de séquence pour l'importation de données	45
Figure 4.1 Architecture globale du système d'acquisition et de contrôle	49
Figure 4.2 Diagramme de déploiement du système d'acquisition et de contrôle.....	51
Figure 4.3 Mécanisme de localisation du système logiciel d'acquisition et de contrôle .	54
Figure 4.4 Fenêtre de dialogue pour l'enregistrement de l'utilisateur	56
Figure 4.5 Fenêtre de dialogue pour la configuration matérielle	57
Figure 4.6 Fenêtre de dialogue pour la configuration du contrôle	59
Figure 4.7 Fenêtre de dialogue pour la configuration des consignes de contrôle	60

Figure 4.8 Fenêtre principale de l'interface graphique de l'application cliente du système DAC	62
Figure 4.9 Architecture orientée-services du système DAC	64
Figure 4.10 Diagramme de paquetage de l'application serveur.....	68
Figure 4.11 Diagramme de classe du paquetage du gestionnaire des modules matériels	69
Figure 4.12 Diagramme de classe du paquetage de services et de leurs interfaces	72
Figure 4.13 Diagramme de paquetage de l'application cliente.....	76
Figure 4.14 Diagramme de classe de l'application cliente et des éléments du paquetage des interfaces de services y étant relatifs	77
Figure 5.1 Boucle de commande par ordinateur adaptée au système DAC thermique ...	80
Figure 5.2 Schéma simplifié représentant la configuration du système physique sur lequel est appliqué le système DAC	81
Figure 5.3 Évolution de la température acquise par le thermocouple du système cylindrique métallique contrôlée avec un contrôleur proportionnel avec gain critique	83
Figure 5.4 Agrandissement d'une zone de la figure 5.3 pour l'estimation de la période critique P_c	83
Figure 5.5 Évolution de la température acquise par le thermocouple du système cylindrique métallique contrôlée avec un contrôleur PID configuré selon la méthode de <i>Ziegler-Nichol</i>	84

Figure 5.6 Agrandissement d'une sous-zone de la figure 5.5 pour l'estimation des critères d'évaluation du contrôle.	85
Figure 5.7 Configuration du moule étudié avec contenant la résine thermodurcissable .	87
Figure 5.8 Évolution de la température et du flux thermique en fonction du temps pour un essai de polymérisation d'une résine vinylester	89
Figure 5.9 Évolution de la température et du flux thermique en fonction du temps pour le chauffage d'une résine vinylester polymérisée.	90
Figure 5.10 Évolution du flux thermique de polymérisation après filtrage en fonction du temps pour une résine vinylester	91
Figure 5.11 Évolution du degré de polymérisation dans le temps pour une résine vinylester	93

LISTE DES ANNEXES

ANNEXE A : Diagramme de cas d'utilisation de système logiciel d'acquisition et de
contrôle 100

ANNEXE B : Spécifications et schéma du Microcontrôleur ADuC845[22] 101

NOTATIONS

$e(\infty)$	Erreur en régime permanent
e_{\max}	Dépassement
K_s	Constance de Seebeck
K_c	Gain proportionnel critique
K_d	Gain dérivatif du P.I.D
K_i	Gain integral du P.I.D
K_p	Gain proportionnel du P.I.D
P_c	Période critique
T_p	Temps de dépassement
T_s	Temps de réponse
A/N	Conversion analogique à numérique
API	Application Programming Interface
CLR	Common Language Runtime
CORBA	Common Object Request Broker Architecture
DNS	Domain Name System

DAC	Data Acquisition and Control
HTTP	Hyper Text Transfer protocol
JRE	Java Runtime Environment
MLI	Modulation par Longueur d'Impulsion
N/A	Conversion numérique à analogique
PID	Proportional Integral Derivative
PC	Personal Computer
RMI	Remote Method Invocation
SCADA	Supervisory Control And Data Acquisition
UML	Unified Modeling Language

INTRODUCTION

À l'heure actuelle, les systèmes d'acquisition et de contrôle interviennent de manière significative au niveau d'une proportion importante de secteurs industriels. En effet, le besoin constant de superviser des systèmes et des procédés prenant part à une certaine industrie nécessite le recours à des outils d'acquisition de données. Aussi la tendance continue en matière d'automatisation des ressources matérielles opérantes s'apparente à une utilisation croissante des systèmes de contrôle.

Un exemple pertinent de ce type d'industries est la fabrication des matériaux composites. Celle-ci connaît un essor impressionnant et continu se justifiant par la panoplie d'avantages qu'elle offre le recours à ce type de technologies et les solutions qu'elle met à disposition des industriels dans le but de répondre à de maints défis techniques. Actuellement, sa popularité s'étend même aux secteurs de technologie avancée en l'occurrence de l'industrie aéronautique, laquelle impose des normes de qualité hautement contraignantes touchant à la fois aux produits et aux processus de fabrication.

Le caractère multidisciplinaire de l'industrie des composites nécessite l'interaction fréquente de multiples acteurs industriels de spécialités variées. Cela entraîne un besoin grandissant de procédures de collaboration efficace et de supervision répartie de procédés variés et mutuellement délocalisés. Un tel objectif ne peut prendre forme que par le biais d'outils informatiques fiables mettant en place des solutions appropriées allant dans ce sens.

D'autre part, la fiabilité croissante des infrastructures de communication et de transmission de données telles que les réseaux informatiques en général, et internet en particulier, représente un point d'intérêt pour les ressources matérielles relatives aux

industries. Aussi, le niveau de performance qu'offre ce type d'infrastructures incite à leur introduction au sein des développements de systèmes critiques.

À la lumière des constats cités précédemment, il existe donc une nécessité et un avantage réel de joindre les technologies de répartition des systèmes à la conception des outils d'acquisition et de contrôle. Toutefois, la performance de ce type de systèmes risque de décroître considérablement par rapport aux possibilités offertes par les systèmes d'acquisition et de contrôle classiques.

Le présent projet a pour but de vérifier l'hypothèse de la possibilité de développer des systèmes d'acquisition de données et de contrôle de procédés répartis à travers les réseaux informatiques. Il se fixe aussi comme objectif l'application de ce type de systèmes à la caractérisation thermique des résines thermodurcissables utilisées dans la fabrication des matériaux composites.

Ce document présente un résumé des principales notions relatives à la conception et au développement d'un système d'acquisition et de contrôle à distance, appliqué à la fabrication des matériaux composites. Tout d'abord, une brève introduction est fournie sur les technologies de répartition des systèmes et des outils d'acquisition et de contrôle. Aussi, une courte récapitulation est donnée relativement aux principales avancées technologiques actuelles dans le domaine traité. Par la suite, une description détaillée est fournie sur les divers aspects portant sur l'analyse et la conception du système développé. Cette description inclut les parties logicielle et matérielle des principales composantes du système. Les résultats de validation du système global sont présentés pour l'application considérée, soit la caractérisation thermique des résines thermodurcissables. Enfin, le document se termine par une conclusion qui résume les points saillants de cette étude.

CHAPITRE 1 : REVUE BIBLIOGRAPHIQUE

Le présent chapitre introduit aux divers concepts et technologies auxquels se rapporte à la conception et le développement des systèmes répartis d'acquisition et de contrôle. Il s'agit d'abord de présenter les enjeux que présente la répartition d'un système industriel sur un réseau informatique. Aussi, seront abordées les considérations à prendre en compte lors de la réalisation d'un système industriel d'acquisition de données. Par la suite, une revue des méthodologies les plus en vigueur pour le contrôle des procédés industriels clôturera ce chapitre.

1.1. Systèmes répartis sur réseaux informatiques

Un système réparti sur réseaux informatiques se définit comme étant un ensemble de composantes logicielles ou matérielles réparties sur plusieurs machines connectées sur un ou plusieurs réseaux informatiques raccordés. Ces composantes inter-communiquent et coordonnent leurs interactions uniquement par transmission de messages [1].

1.1.1. Avantages de systèmes répartis

Les systèmes répartis sur réseaux présentent un grand nombre de possibilités intéressantes par opposition aux systèmes centralisés. Ces avantages s'énoncent entre autres comme suit.

- La possibilité de redondance des ressources donne une disponibilité accrue et une meilleure tolérance aux défaillances [3].
- Un partage optimisé des ressources informatiques est facilement réalisable [1].

- Une capacité importante d'évolution est permise [3].
- Sur le plan économique, les systèmes répartis offrent un rapport performance/prix meilleur que celui qu'offrent les systèmes centralisés [2].

1.1.2. Architectures de systèmes répartis

L'architecture d'un système réparti se rapporte à l'organisation de ses composantes, aux relations qui existent entre elles, à leur environnement d'exécution et aux règles qui régissent leurs interactions mutuelles [1, 4].

La conception d'une architecture de système réparti est fondée en général sur deux modèles principaux, à savoir les modèles pair à pair et client/serveur. Cette classification est faite en fonction des responsabilités de chaque composante au sein du système.

Le modèle pair à pair

Ce modèle stipule que toutes les composantes distribuées jouent un rôle commun, dans le sens où chaque nœud du système offre des services similaires. Ainsi, tous les éléments du système global bénéficient des mêmes droits d'accès à aux services disponibles. La figure 1.1 expose le concept d'une architecture pair à pair.

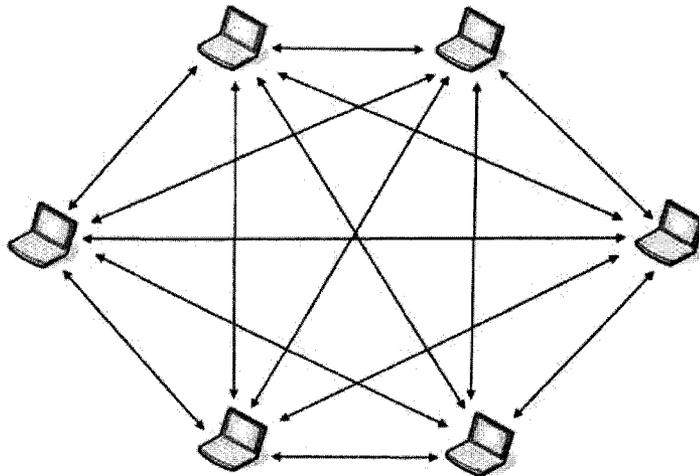


Figure 1.1 Exemple d'architecture pair à pair

Le schéma de la figure 1.1 illustre les diverses interactions possibles entre les éléments du système réparti. Cela correspond à toutes les combinaisons paires possibles des composantes du système. Cette architecture présente les avantages suivants :

- partage équitable et maximale des ressources entre les composantes du système en termes de bande passante, d'espace de stockage et de puissance de calcul ;
- disponibilité accrue des services grâce à la redondance des éléments qui les hébergent ;
- architecture peu coûteuse en termes de déploiement initial.

Toutefois, une telle architecture présente un ensemble d'inconvénients qui limitent son utilisation répandue. Ceux-là se présentent comme suit.

- Le manque de centralisation des données réduit considérablement la possibilité d'une manipulation partagée de ces données.
- La complexité reste difficilement gérable au niveau de l'administration du système comportant un nombre élevé de nœuds.

Le modèle client/serveur

Ce modèle définit des rôles différents pour les composantes du système réparti. Certains éléments jouent le rôle de *serveur* et ont la responsabilité d'offrir un nombre de services et de ressources à d'autres composantes consommatrices. De telles composantes sont nommées *clients*. La figure 1.2 montre un exemple d'une architecture client/serveur :

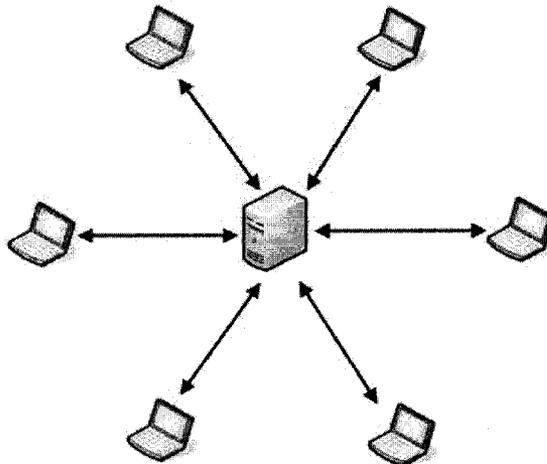


Figure 1.2 Exemple d'architecture client/serveur

Les avantages qu'offre une telle approche se présentent comme suit.

- Une gestion centralisée des ressources et des services reste accessible. Elle a trait aux mises à jour, au contrôle d'accès et à la sécurisation des données. Il reste ainsi possible d'établir des procédures efficaces d'exploitation collaborative de ces ressources et services.
- La maintenabilité du système se concentre en général sur des points de fonctionnement bien dénombrables. Cela correspond en général aux éléments de type serveur.

Ce type d'architecture possède néanmoins quelques désavantages qui s'énumèrent comme suit.

- Un risque de saturation des services qu'offrent les éléments serveurs existe dans le cas d'un nombre important d'éléments clients en exploitation simultanée de ces services.
- Une disponibilité réduite des services. En effet, les points de défaillance dénombrable du système se situent au niveau des éléments serveur.
- Les coûts de déploiement sont plus élevés en comparaison avec ceux des architectures de type pair à pair.

Les modèles d'architecture présentés peuvent être combinés dans le but d'établir des architectures hybrides palliant aux problèmes que possèdent chacun des modèles singulièrement. En effet, il est possible d'imaginer des architectures combinant plusieurs serveurs spécialisés, dont chacun possède d'autres images identiques afin de balancer les charges de service.

Mentionnons que les aspects client et serveur demeurent des rôles que peuvent jouer les composantes d'un système réparti. Cela dit, une même composante peut aussi combiner les deux rôles dans le cadre d'architectures hybrides évoluées. La figure 1.3 illustre l'architecture hybride d'un système de partage de fichiers :

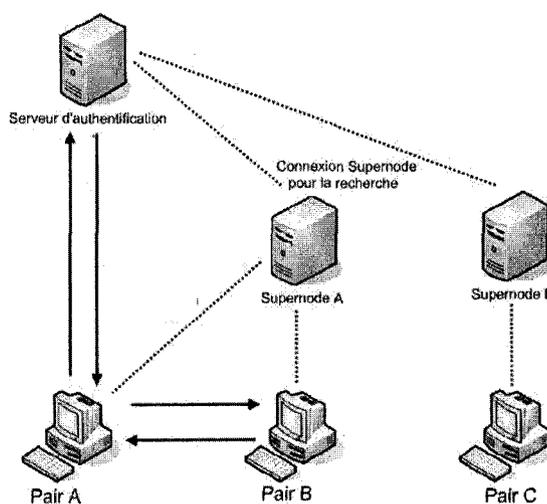


Figure 1.3 Architecture hybride d'un système de partage de fichiers

1.1.3. Technologies de répartition des systèmes

L'élaboration d'un système réparti consiste essentiellement en l'intégration de fonctionnalités d'échange de messages, de synchronisation et de partage de ressources entre les diverses composantes du système. À cette fin, il existe plusieurs mécanismes de communication interprocessus qui permettent de concrétiser cet aspect de la conception.

1.1.3.1.Socket API

Le concept de socket fait référence à un point d'accès local au niveau d'une machine connectée au réseau informatique. Ce point d'accès est géré par le système d'exploitation et est identifié par un doublet constitué de l'adresse IP de la machine et d'un numéro de port unique. Cela est nécessaire à son accessibilité à distance [10]. Ce mécanisme offre une interface de programmation (API) pour l'intégration de fonctionnalités de communication interprocessus à travers le réseau. Toutefois, cette API présente certaines complexités :

- L'API est considérablement limitée quant aux fonctionnalités qu'elle encapsule, dans la mesure où un nombre réduit de fonctions est utilisé pour la réalisation de tous les aspects supportés. Cela dit, l'usage d'une telle interface est considérablement complexe.
- L'API des sockets reste étroitement dépendante des systèmes d'exploitation hautement hétérogènes. Cela en réduit amplement la portabilité.

De tels constats limitent les possibilités de conception de logiciels de grande envergure par utilisation directe à l'API des sockets.

1.1.3.2.Intergiciels orientés-objets

Un intergiciel orienté-objets constitue une couche logicielle sous-jacente aux applications logicielles dont la responsabilité est d'assurer et les communications entre ces dernières. Le but d'une telle couche logicielle est de garantir une transparence accrue des détails de communication interprocessus assurés par les systèmes d'exploitation par rapport aux applications communicantes [3]. Cela dit, l'intergiciel répond aux difficultés

posées par l'utilisation des mécanismes de bas niveau offerts par les systèmes d'exploitation, à savoir l'API des sockets.

Il existe divers types d'intergiciels qui s'adaptent chacun à la panoplie d'architectures logicielles qui existent. La présente étude met l'emphase sur le type d'intergiciel orienté-objet. Techniquement parlant, les principales tâches remplies par ce dernier se résument comme suit :

- *Support du modèle objet.* Cela permet d'intégrer le concept d'objet logiciel localisé à distance et transférable entre les applications communicantes.
- *Interaction opérationnelle.* L'interaction inter-objets doit être accessible à travers l'intergiciel. Concrètement, un objet doit être capable d'invoquer les méthodes d'un autre objet se trouvant sur une machine localisée à distance.
- *Transparence de la répartition.* D'un point de vue applicatif, l'invocation d'un objet localisé à distance est réalisée de manière similaire à celle réservée aux objets locaux.
- *Indépendance technologique.* L'intergiciel supporte l'intégration de technologies hétérogènes [3].

La figure 1.4 présente l'architecture typique d'un système réparti fondé sur un intergiciel orienté-objets :

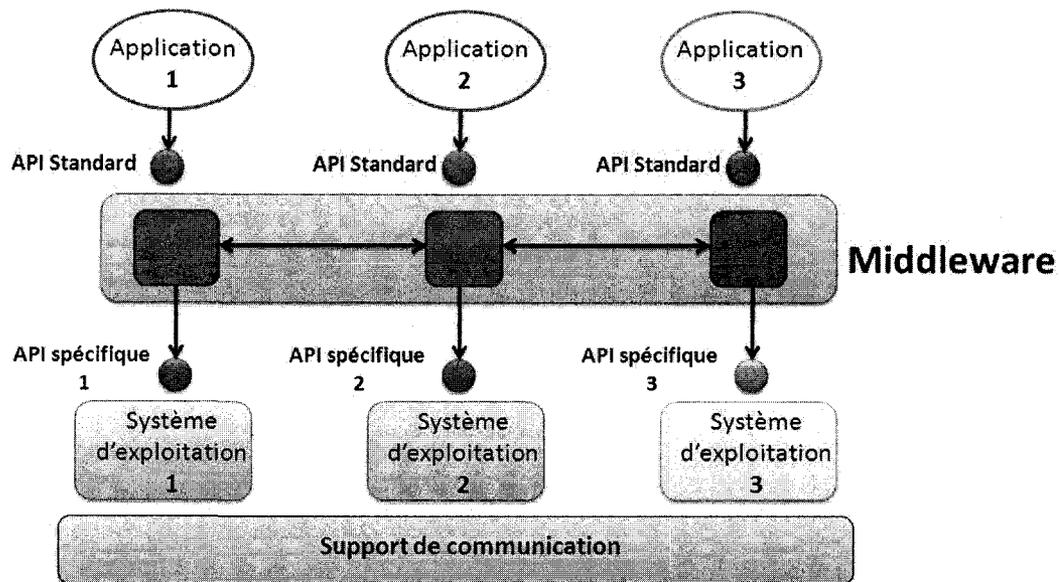


Figure 1.4 Exemple d'architecture de système réparti fondé sur un intergiciel orienté-objets [7]

Le schéma de la figure 1.4 explicite l'avantage qu'apporte une architecture fondée sur un intergiciel, à savoir l'unicité des API de développement. En effet, l'interface de développement qu'offre l'intégiciel reste indépendante de la nature de la plateforme sur laquelle s'exécute l'application. Cela va en adéquation avec l'une des raisons d'être principale des intergiciels, en ce qui a trait à la transparence de l'hétérogénéité des plateformes par rapport aux applications qui s'y exécutent.

Tel que mentionné, un système réparti bâti sur un intergiciel orienté-objets est composé d'instances (ou objets) logicielles invocables à distance, nommées services. Il existe plusieurs approches de gestion de ces objets en termes de leur cycle de vie. Ce dernier correspond à la durée qui s'écoule entre le moment de la création de l'objet et le moment de sa destruction. Les approches de gestion de ces objets font partie intégrante de la

conception de tout intergiciel orienté-objets. Leur diversité donne la possibilité d'optimiser la conception du système réparti en fonction des requis fonctionnels auxquels il est censé répondre. Il est possible de distinguer trois choix possibles pour la gestion de cette durée de vie :

- *Instances statiques.* Elles correspondent à des objets statiques dont la durée de vie est celle de l'application qui les héberge. La création de tels objets est pertinente dans le cas d'une nécessité d'avoir des objets avec des états accessibles en permanence.
- *Instance sur requête.* Le service est créé à la réception d'une requête dans le but d'y répondre. La destruction du service suit l'envoi de la réponse. Cela paraît utile dans le cas où l'état du service est peu pertinent à l'application. Une telle méthode présente l'avantage d'assurer une meilleure gestion des ressources de l'application qui fournit les services.
- *Instance par client.* Une instance du service est créée afin de servir un client en particulier. La destruction du service est initiée à la suite de la déconnexion du client.

Dans ce qui suit seront introduits brièvement les principaux intergiciels orientés-objets disponibles dans le marché des systèmes répartis.

CORBA

Il s'agit d'un standard architectural destiné à la réalisation d'applications orientées-objets réparties. Cette réalisation est indépendante du langage de développement utilisé et des plateformes sur lesquelles s'exécutent les applications [5]. Ce standard d'intergiciel a la particularité de fournir un nombre de services préétablis pour assurer la réalisation de plusieurs fonctionnalités communément rencontrées au niveau des applications réparties. La complexité du standard reste toutefois non négligeable en termes de manipulation et d'assimilation. Aussi, rares sont les implémentations ouvertes et complètes de ce standard.

JAVA RMI

Cet intergiciel réalisé par la firme *Sun* permet à un objet instancié et géré par une machine virtuelle Java (*JRE*), d'invoquer d'autres objets hébergés par d'autres machines virtuelles Java [8]. Ces dernières représentent les plateformes sous-jacentes aux applications développées en langage Java. Cela dit, cet intergiciel tire profit de tous les avantages qu'offre cette plateforme, soit la prise en charge des systèmes d'exploitation hétérogènes. Cependant, il reste limité aux applications développées en langage Java.

.Net Remoting

Il s'agit d'un intergiciel développé par la firme *Microsoft* dans le cadre de la plateforme *.Net*. Similairement à *Java RMI*, cet intergiciel offre des possibilités d'invocations d'objets s'exécutant sur des machines virtuelles *CLR* localisées à distance. Il s'agit des machines virtuelles introduites dans le standard *.Net*. L'avantage d'une telle plateforme par rapport à la *JRE* réside dans la possibilité d'intégrer des modules implémentés dans des langages différents. Un autre avantage réside dans l'introduction du concept de

machine virtuelle qui reste absente de CORBA. En effet, le choix de la solution .Net Remoting permet d'exploiter les mécanismes efficaces et avancés de la plateforme *CLR* pour la gestion des ressources applicatives. À cela s'ajoutent d'autres avantages pratiques qui s'énumèrent comme suit [9] :

- une facilité accrue au niveau de l'implémentation des aspects de communication à distance ;
- une architecture extensible dans le sens où les protocoles et les procédures implémentés au sein de l'intergiciel peuvent être étendus à de nouvelles approches conçues en fonction des besoins applicatifs ;
- les interfaces des objets services correspondent aussi aux interfaces utilisées du côté des applications exploitantes et s'implémentent de manière similaire au mode habituel de développement des applications ;
- la possibilité d'échange mutuel d'objets complexes entre les applications communicantes.

1.2. Systèmes d'acquisition de données

L'acquisition de données consiste en la capture de signaux représentatifs d'un phénomène physique réel par usage de sources de mesure [13]. Cela regroupe aussi la numérisation de ces signaux pour des fins de stockage, d'analyse et de représentation graphique [14]. Les signaux extraits sont exploités dans l'étude du phénomène physique correspondant. Ils interviennent aussi dans l'analyse et le suivi du phénomène. Une application courante de l'acquisition de données est le contrôle de procédés. Cela consiste en le maintien de l'état ou des états d'un phénomène donné en concordance à certaines consignes imposées.

L'exploitation des signaux s'effectue en général au moyen de systèmes électriques et informatiques. Le canal d'acquisition des données est constitué en général de plusieurs éléments comme les multiplexeurs et les convertisseurs analogique à numérique. D'autres éléments interviennent pour le transfert de données aux unités de traitement en l'occurrence les ordinateurs. La figure 1.5 illustre le diagramme d'un système d'acquisition typique.

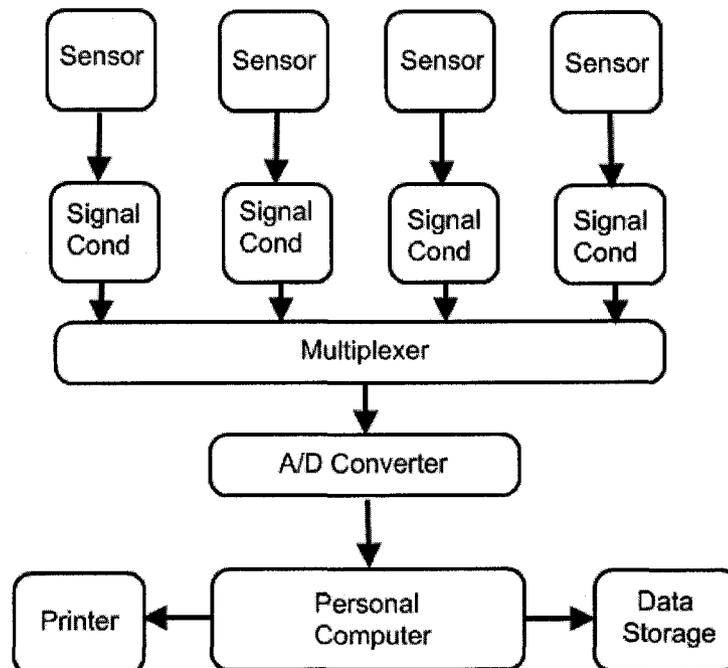


Figure 1.5 Diagramme d'un système d'acquisition [16]

1.2.1. Éléments matériels de systèmes d'acquisition

Plusieurs éléments interviennent dans le processus d'acquisition de données. L'élément premier du canal d'acquisition est le capteur. Il s'agit d'une composante matérielle dont le rôle est de convertir un phénomène physique en signal électrique mesurable. La fiabilité des données fournies par les systèmes d'acquisition dépend étroitement de la capacité des capteurs à reproduire l'état du phénomène physique étudié sous forme d'un signal électrique [14].

Un exemple populaire de capteur est le thermocouple. Ce dernier opère selon l'effet thermoélectrique **Seebeck** en fonction des matériaux qui le composent dans le but de produire un voltage en fonction de la température de ces derniers.

Un autre élément matériel à l'impact majeur sur le fonctionnement du système d'acquisition est le convertisseur analogique numérique. Cet élément a pour but de transformer le signal électrique analogique transmis par le capteur en signal numérique intelligible par l'unité de traitement. Plusieurs paramètres agissent sur la fiabilité des résultats fournis à la suite de la conversion analogique numérique :

- La fréquence d'échantillonnage du signal analogique est fondée sur la loi de **Nyquist** [23], qui stipule que la reproduction d'un signal analogique, après sa conversion en signal numérique, n'est possible que lorsque la fréquence à laquelle il est échantillonné est supérieure ou égale au double de la fréquence la plus élevée des harmoniques du signal.
- La plage de conversion constitue aussi un élément important à prendre en compte pour évaluer la concordance du convertisseur avec ses conditions d'utilisation. Dans le cas idéal, cette plage doit couvrir tout l'intervalle des valeurs potentielles du signal analogique sans aucun dépassement. En pratique, il est préférable de choisir les plages de conversion qui s'approchent au mieux de la plage idéale. Cela suppose une

connaissance préalable des valeurs que peut prendre le signal étudié. Des plages insuffisantes ou excessivement larges induisent la perte de certains aspects du signal capté.

- La résolution du convertisseur possède un impact majeur sur la qualité de la conversion. Cela réfère au nombre de bits sur lesquels est codé le signal converti. Il faut signaler que la précision de la conversion s'accroît avec le nombre de bits considéré.

L'intervention des multiplexeurs dans le processus de conversion de données est pertinente dans le cas du traitement de plusieurs signaux à travers des composantes communes. Un tel élément permet de transmettre des données différentes suivant un canal commun pour l'acquisition des données, en excluant le capteur du signal. Enfin, une phase de conditionnement [14] est appliquée au signal dans les systèmes d'acquisition. En effet, les signaux captés sont souvent filtrés afin d'obtenir des signaux lisses.

1.2.2. Logiciels d'acquisition de données

L'exploitation des données récupérées s'effectue par le biais de logiciels s'exécutant sur des machines connectées aux modules matériels en charge de l'acquisition. Les capacités de ces logiciels peuvent être limitées au traitement des données acquises à une étape ultérieure au processus d'acquisition. Cela s'effectue par chargement des données à partir d'un support de stockage. Aussi il existe des logiciels ayant la capacité de procéder à l'acquisition en temps réel dans le sens où les traitements réservés aux données sont effectués parallèlement au processus d'acquisition lui-même. Ces traitements consistent entre autres à la représentation graphique des données acquises, à leur stockage ou encore à leur analyse.

Les logiciels destinés à l'acquisition de données peuvent avoir la spécificité d'être d'usage général et indépendant de la nature des données collectées. Cela est d'autant plus vrai dans le cas de traitements différés des données acquises. Il s'agit en général de logiciels offrant des outils communs de traitement de signal. Par opposition à cette catégorie de logiciels, il existe une deuxième famille de logiciels, permettant des traitements spécifiques et en rapport direct avec la nature du signal capté. Ce type de logiciels est mieux adapté à l'acquisition en temps-réel, puisqu'ils prennent en considération les détails techniques relatifs à la communication avec le module matériel d'acquisition de données.

1.3. Commande de procédés par ordinateur

Tel que mentionné auparavant, une application pertinente aux données issues du processus d'acquisition s'avère être le contrôle de procédés. Cela consiste à maintenir un état caractéristique du procédé en fonction de consignes imposées. À cette fin, il est nécessaire d'effectuer l'acquisition des données représentatives de cet état afin de faire le suivi de ce dernier et de prendre les décisions nécessaires pour le maintien de la conformité aux consignes prédéfinies. Une telle procédure peut être schématisée selon la boucle de la figure 1.6. Celle-ci illustre le niveau d'intervention du système d'acquisition au sein de la boucle de contrôle. En effet, les données acquises constituent des points d'entrée pour l'unité de contrôle dont le rôle est de générer des commandes permettant d'atteindre les consignes préétablies.

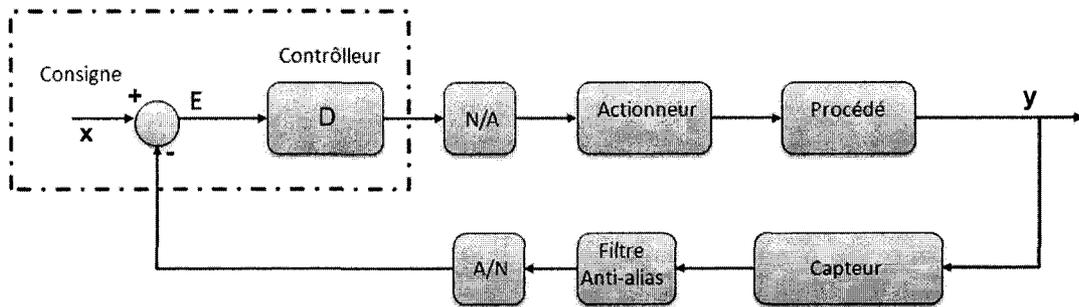


Figure 1.6 Boucle typique de commande de procédé par ordinateur

La commande de procédés par ordinateur prend en considération l'aspect discrétisé des signaux d'entrées. À cette fin, l'ensemble des éléments du système sont modélisés sous forme d'équations aux différences finies:

$$y(nT) = \sum_{i=0}^r L_i x(nT - iT) - \sum_{i=1}^m K_i y(nT - iT) \quad (1.1)$$

où T réfère à la période d'échantillonnage, n est le nombre d'échantillons, $y(t)$ la sortie du système à l'instant t et $x(t)$ la consigne du système. Les paramètres L_i et K_i représentent des coefficients dépendants du contrôle.

Cette équation stipule que la sortie du procédé à l'instant nT est fonction de la somme des sorties pendant un certain nombre de périodes d'acquisition précédentes et de la somme de ses états passés.

La qualité du contrôle s'évalue en général en fonction de plusieurs critères se classifiant en deux catégories principales. La première catégorie correspond au régime transitoire du système qui se définit comme l'étape du cycle de contrôle durant laquelle l'état contrôlé ne présente pas de stabilité. Ces critères s'évaluent dans le cas d'une consigne fixe en forme d'échelon. De tels critères s'énumèrent comme suit :

- Le dépassement e_{max} . Il s'agit de la différence entre les valeurs maximale et finale que prend l'état du système.
- Le temps de dépassement T_p . Cela correspond à l'instant de temps auquel l'état du système atteint sa valeur maximale.
- Le temps de réponse à 2% T_s . cette valeur se confond avec le premier instant de temps auquel l'état du système atteint 2% de la consigne imposée.

Un deuxième critère d'évaluation du contrôle se rapporte à l'erreur du système en régime permanent. Celle-ci se définit par la relation suivante :

$$\lim_{t \rightarrow \infty} e(t) = 0 \quad (1.3)$$

La figure 1.7 illustre les critères d'évaluation des algorithmes de contrôle. La fonction de transfert du système dépend de la fonction de transfert du contrôleur. Le choix d'une telle fonction s'effectue au moyen d'une variété d'algorithmes. Pour l'algorithme P.I.D par exemple, le calcul de la commande à la sortie du contrôleur à un instant donné t est fait en fonction de l'erreur par rapport à la consigne au même instant t . Des approximations de l'intégral de cette erreur dans le temps et de sa dérivée par rapport au temps sont à la base de la détermination de la commande. L'expression de la commande pour une équation aux différences finies peut prendre la forme suivante :

$$M(nT) = K_p E(nT) + K_i T \sum_{i=0}^{n-1} E(kT) + K_d \frac{E(nT) - E((n-1)T)}{T} \quad (1.4)$$

où M représente la commande de contrôle et E l'erreur de contrôle. Notons que la qualité du contrôle dépend du choix des facteurs K_p , K_i , K_d , appelés aussi gains. Il existe plusieurs stratégies pour optimiser de ces gains. Dans le cas où le modèle du système est connu, la stratégie de placement des pôles s'avère être une approche pertinente pour l'évaluation de gains optimaux du contrôleur. Il existe aussi des méthodes empiriques

permettant de procéder à l'optimisation des gains dans le cas où le modèle du phénomène est inconnu, en l'occurrence la méthode de **Ziegler Nichol** [17]. Plus Des explications plus détaillées seront fournies au chapitre 5 sur cette méthode. Il existe aussi divers contrôleurs plus spécialisés dont les détails seront omis dans le cadre de l'étude présente. Il s'agit entre autres des contrôleurs à réponse pile, le prédicateur de Smith, le contrôleur par inférence.

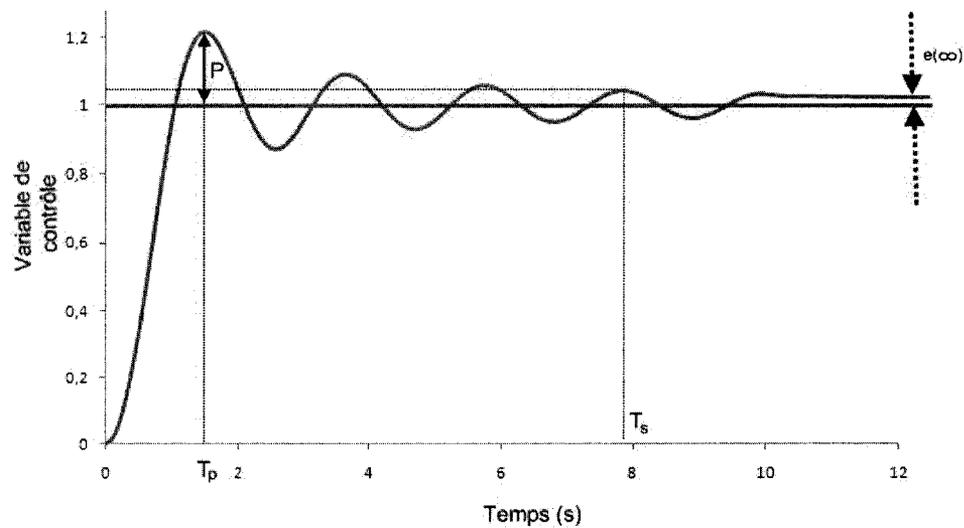


Figure 1.7 Critères d'évaluation du contrôle des systèmes

1.4. Tendances des systèmes d'acquisition de données et de contrôle

Les systèmes de contrôle et d'acquisition de données sont abondamment utilisés dans le secteur industriel afin de sécuriser les processus industriels. À l'heure actuelle, l'évolution de ce genre de systèmes prend diverses tendances. En effet, la fiabilité des plateformes de communication actuelles permet d'introduire plus efficacement plusieurs aspects nouveaux, en l'occurrence les accès à distance aux données ainsi que la concurrence et le partage de leur exploitation. Cela consiste en l'établissement

d'architectures composées de nœuds centraux d'acquisition de données, responsables de la distribution de l'information entre plusieurs unités consommatrices localisées à des emplacements différents à travers le réseau [19].

L'exploitation des réseaux informatiques dans le domaine de l'acquisition des données entraîne des implications notables sur les processus de conception et de développement des logiciels dédiés à ces fins. De nouvelles considérations doivent être introduites pour permettre une gestion efficace du partage des données sans pour autant réduire les performances des fonctionnalités de base.

Plusieurs efforts répondent à cette vision des systèmes d'acquisition et de contrôle. En effet, Zhu [20] a contribué à cette tendance en introduisant un module logiciel dédié aux applications d'acquisition de données et de contrôle en réseau. Il s'agit d'une solution modulaire généralisable ayant été validée, entre autres, pour le suivi d'un réseau de capteurs sans-fil. D'autre part, R. Arnaudov et I. Dochev [21] ont réalisé un système d'acquisition contrôlé à partir du réseau internet et utilisant les services et les technologies web pour cette fin. Le web constitue l'un des outils les plus populaires offrant la possibilité de partage et de manipulation à distance des données [6]. Il s'agit en effet d'un réseau virtuel bâti sur internet et qui permet à plusieurs éléments clients et serveurs d'inter-communiquer par le biais du protocole HTTP [12]. Ce dernier fait référence au protocole d'échange de pages web communément utilisées par les usagers des services internet.

Il existe plusieurs outils commerciaux qui vont dans le sens de cette vision d'outils collaboratifs pour le suivi industriel. Plusieurs d'entre eux interviennent dans le domaine de la conception assistée par ordinateur. En effet, les outils *Centric Software Pivotal Studio*, *RealityWave ConceptStation* et *Autodesk StreamLine* offrent la possibilité de visualiser en mode partagé des modèles géométriques et de les manipuler. Ils regroupent aussi des fonctionnalités d'échange de messages [19].

Deux paradigmes sont majoritairement utilisés pour la conception des outils collaboratifs. Le premier focalise sur le développement du côté client dans le système réparti et fait appel à des technologies comme Java Applet et Ms ActiveX [6]. Il s'agit en effet de modules logiciels téléchargeables automatiquement par des applications clientes comme les fureteurs web afin de prendre en charge les aspects de partage réparti des données. Le deuxième paradigme consiste en un modèle hybride concentrant les efforts de développement à la fois sur les composantes clients et serveurs. Les systèmes fondés sur un tel paradigme de développement font appel à des technologies serveur comme Java RMI et .Net Remoting.

CHAPITRE 2 : SYSTÈME MATÉRIEL D'ACQUISITION ET DE CONTRÔLE THERMIQUES

Le présent chapitre a pour objectif de présenter le système matériel d'acquisition et de contrôle thermique *CureLab*. Cela comprend la description des unités principales qui composent le système et les interactions qui existent entre elles. Les mécanismes de communication et d'intégration des deux parties matérielle et logicielle du système seront explicités.

2.1. Architecture du système d'acquisition et de contrôle thermique

Un système d'acquisition et de contrôle (DAC) thermique a pour objectif de faire le suivi d'un procédé à travers la mesure de données thermiques. Le procédé peut ainsi être contrôlé sur la base des données acquises. Ces données thermiques sont les températures à des positions du système physique et les flux thermiques traversant des sous-sections dudit système physique. Les figures 2.1 et 2.2 montrent des photographies des composantes du système matériel DAC.

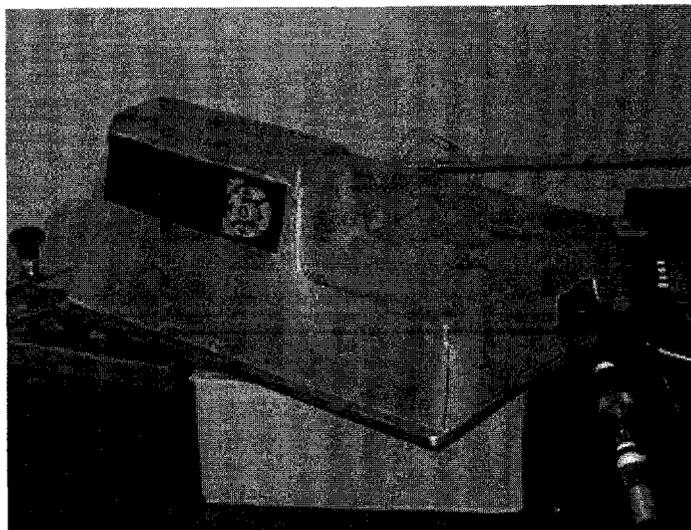
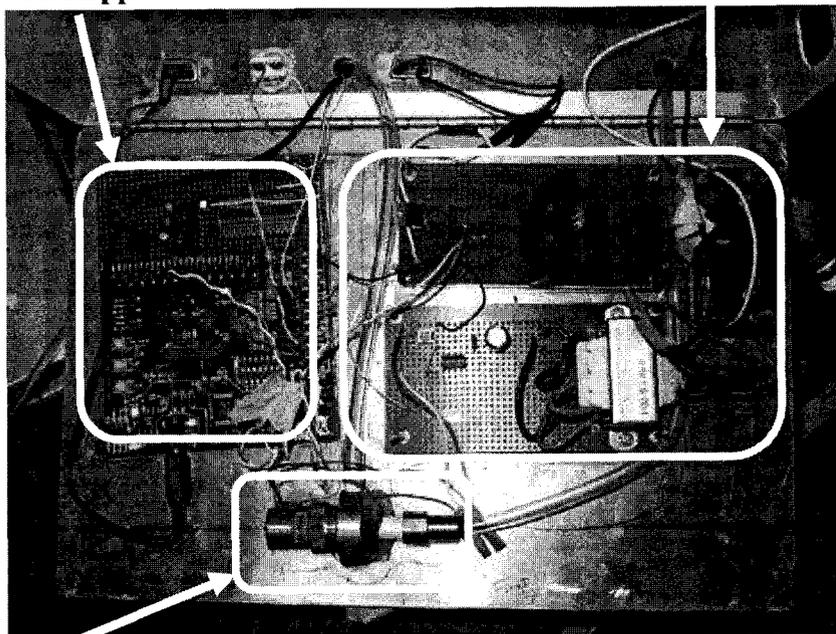


Figure 2.1 Vue externe du système matériel DAC

**Microcontrôleur +
Carte de développement**

**Circuit
d'alimentation**



**Valve pour le
refroidissement**

Figure 2.2 Vue interne du système matériel DAC

La figure 2.3 illustre l'architecture matérielle globale du système DAC thermique :

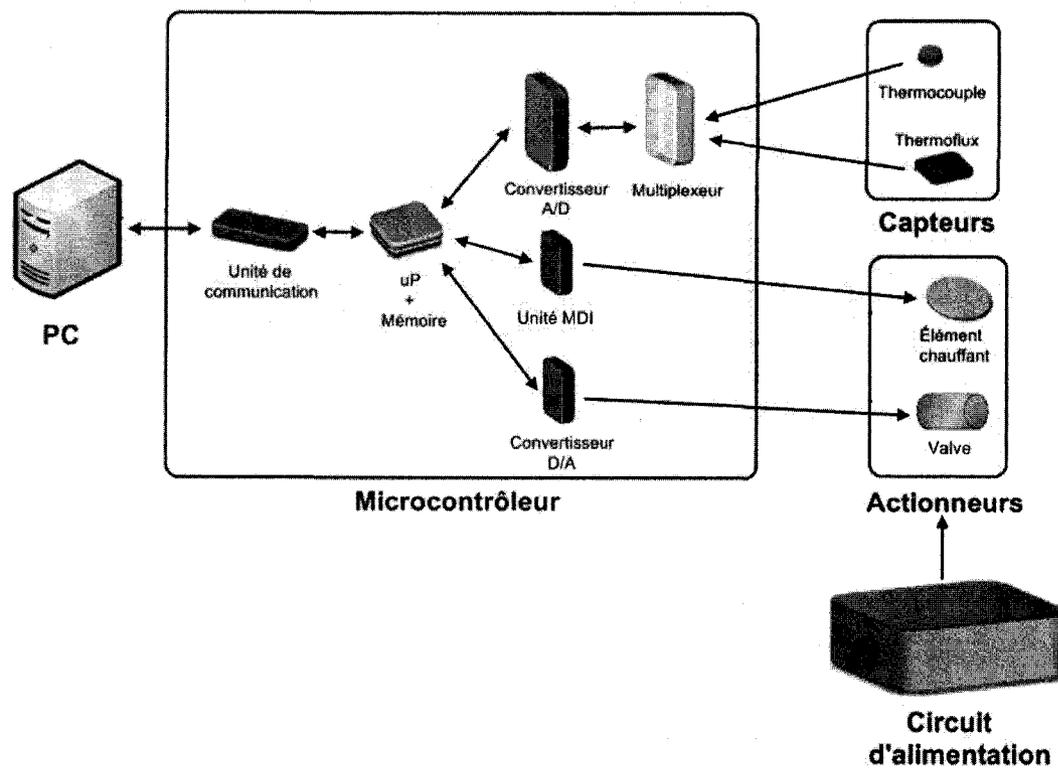


Figure 2.3 Architecture matérielle globale du système DAC

2.1.1. Capteurs

La température à une position donnée peut être mesurée au moyen de thermocouples. La mesure s'effectue à travers du courant électrique induit aux bornes du capteur. L'induction de ce courant est due à l'effet thermoélectrique **Seebeck** qui dépend des propriétés des matériaux qui composent le thermocouple. Le courant induit a la propriété d'être proportionnel à la température du capteur. En principe, le capteur est caractérisé

par une réponse qui dépend de la température. La figure 2.4 schématise la structure d'un capteur thermocouple typique.

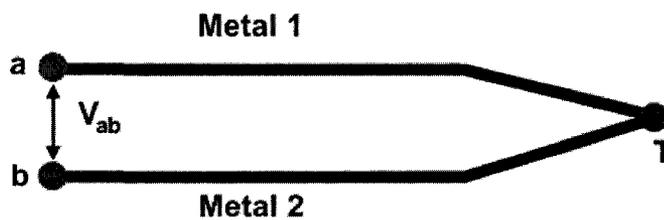


Figure 2.4 Schéma de la structure typique d'un thermocouple

Les capteurs thermoflux servent à mesurer le flux thermique qui traverse une surface donnée. Chaque capteur est constitué d'une série de thermocouples étalés sur les deux faces du capteur. Ces deux faces sont séparées par une barrière thermique, qui induit un gradient de température entre les deux faces au moment où un flux thermique traverse le capteur. Ce gradient est proportionnel à la valeur du flux thermique. Cela toutefois suppose que le flux thermique est constant à travers toute la surface du capteur. La calibration de ce type de capteur présente certaines difficultés dans la mesure où il est délicat de produire des sources de flux thermique stables dont la connaissance préalable permettra d'évaluer la constante de proportionnalité caractéristique du capteur. La figure 2.5 expose la structure d'un capteur de flux thermique typique.

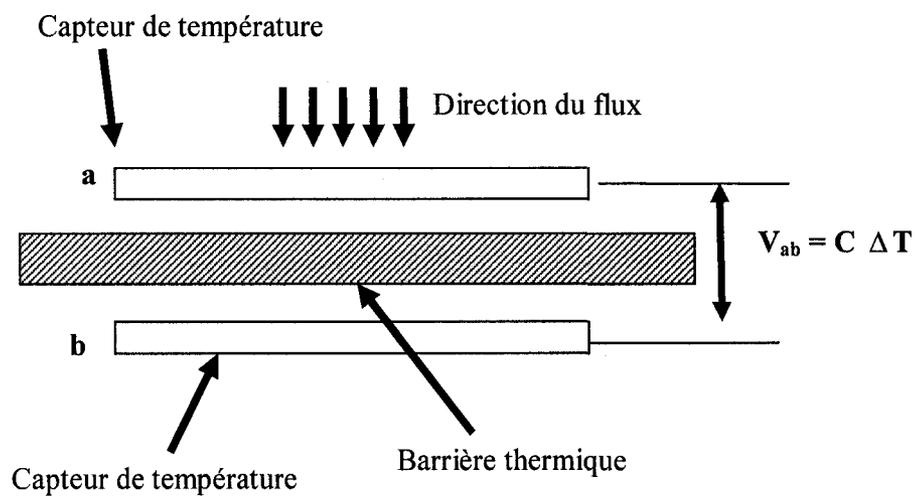


Figure 2.5 Schéma de la structure d'un capteur de flux thermique [24]

La figure 2.6 illustre les capteurs thermocouple et thermoflux utilisés au niveau du système DAC courant.

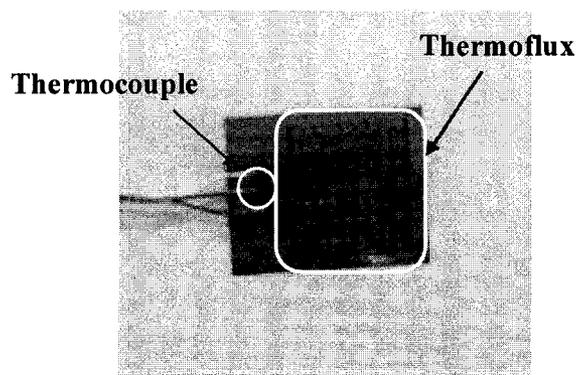


Figure 2.6 Photographie des capteurs thermocouple et thermoflux

Cette figure illustre l'aspect combiné des deux types de capteurs présentés. En effet, un tel capteur, produit par la compagnie **Omega**, permet d'acquérir de manière indépendante la température à une position donnée ainsi que le flux thermique d'une surface comprenant ce point.

2.1.2. Actionneurs

Les actionneurs interviennent au niveau du contrôle du procédé. Il s'agit d'éléments permettant d'appliquer les commandes générées par l'application de l'algorithme de contrôle. Les actionneurs considérés permettent d'agir sur la température du système. Cela peut se faire par génération de chaleur par effet joule, au moyen d'un élément résistif. L'abaissement de la température peut être obtenu par principe de convection en appliquant un flux d'air à une température plus faible. Un exemple d'actionneur permettant de concrétiser cet effet est la valve d'air comprimé. Les figures 2.7 et 2.8 présentent respectivement des photographies de l'élément chauffant et de la valve intégrés au système DAC développé.

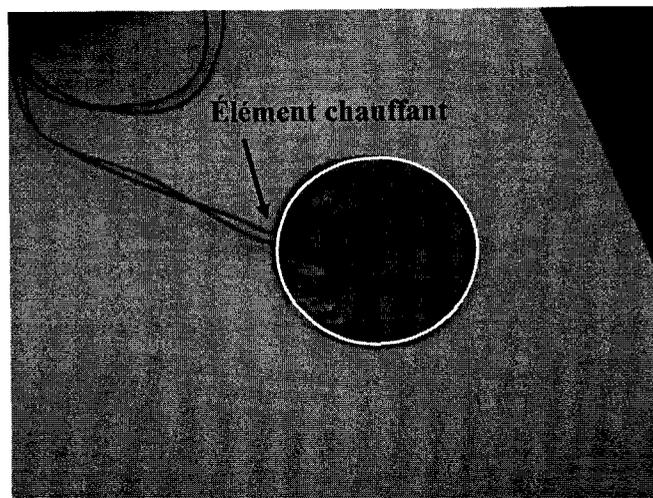


Figure 2.7 Photographie de l'élément chauffant

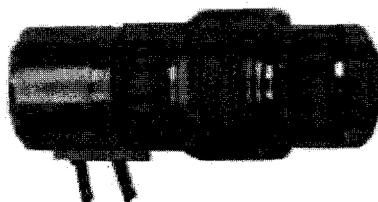


Figure 2.8 Photographie de la valve d'air comprimé

L'élément chauffant de nature résistive produit de la chaleur avec une certaine puissance proportionnelle à la tension appliquée à ses bornes. Cette constante de proportionnalité n'est autre que la valeur de la résistance de l'élément chauffant. Quoique la loi d'Ohm qui régit le comportement de cet élément chauffant n'impose aucune limite sur la puissance maximale pouvant être produite par sa résistance caractéristique, les matériaux

qui le composent ne permettent pas d'opérer au-delà d'une température donnée. La puissance nominale correspondante est de 31.4W.

La valve de refroidissement opère aussi sous tension électrique. Cette dernière a pour effet de faire varier l'ouverture la valve, et par conséquent agit sur le débit induit à la sortie de cette dernière. La figure 2.9 illustre les spécifications techniques du fonctionnement de la valve pour une pression d'air à son entrée de l'ordre de 100 psi. Le rapport entre le courant électrique imposé à l'entrée de la valve de refroidissement et le débit d'air induit à sa sortie peut être approximé selon une relation linéaire, tel qu'illustré à la figure 2.10.

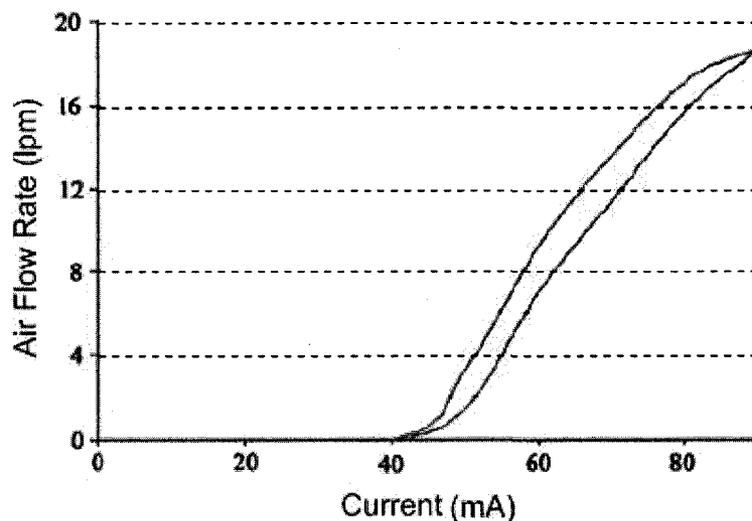


Figure 2.9 Évolution du débit d'air induit à la sortie de la valve en fonction du courant électrique imposé pour une pression d'air à l'entrée de l'ordre de 100 psi.

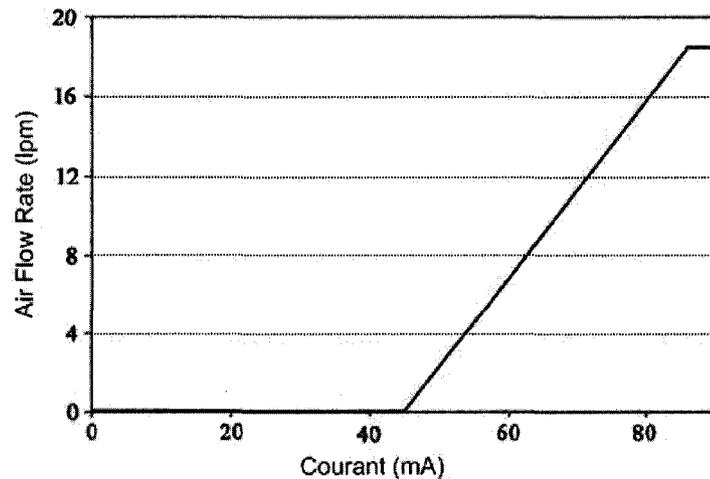


Figure 2.10 Linéarisation du rapport du débit d'air induit à la sortie de la valve et du courant électrique imposé pour une pression d'entrée de l'air de l'ordre de 100 psi.

Une opérabilité fonctionnelle de la valve de refroidissement correspond à sa plage de fonctionnement. Celle-ci se confond avec la partie linéaire de la courbe caractéristique. En effet, la commande minimale théorique imposée à la valve sera de l'ordre de 45 mA alors que sa limite maximale correspondra à un courant d'entrée d'une valeur de 90 mA. La calibration empirique des limites de fonctionnement de la valve a permis de valider la pertinence de ces marges. En effet, la valve présente des points fonctionnels qui convergent vers ses valeurs nominales.

Les besoins énergétiques de chaque actionneur sont générés par le biais d'un circuit d'alimentation analogique opérant avec un courant alternatif de tension égale à 120 V et d'une fréquence de 60 Hz. Ses principales fonctions touchent à la transformation de la

tension alternative à son entrée en tension continue dont la valeur est égale à celle requise par l'actionneur utilisé.

2.1.3. Microcontrôleur

Le microcontrôleur constitue le cœur même de la partie matérielle du système DAC. En effet, il est responsable d'initier l'acquisition des données correspondantes aux capteurs supportés et d'appliquer les commandes relatives au contrôle. Il prend aussi en charge la communication avec la machine serveur. Les échanges d'informations correspondent à la réception des requêtes d'acquisition des données ou d'application de la commande, et à l'envoi des données collectées à la machine serveur. Les principales sous-composantes du microcontrôleur sont les suivantes :

- *Unité de communication*

Celle-ci prend en charge l'envoi et la réception de données numériques à travers le canal de communication supporté. La version actuelle permet de communiquer à travers l'interface RS 232, plus communément connu sous le nom de port série. Il faut se rappeler que ce dernier permet de transférer les unités d'encodage des données de manière successive. Ces unités de communication prennent la forme de bits (0 ou 1). La notification de la réception des données envoyées par la machine interlocutrice est aussi effectuée par l'unité courante.

- *Microprocesseur*

Cette unité se charge de l'exécution du code d'instructions. À chaque cycle d'exécution, une instruction combinée aux données qui s'y rapportent est envoyée à l'entrée du microprocesseur. L'exécution de l'instruction donne en sortie des données qui seront

enregistrées au niveau de la mémoire de données ou au sein d'emplacements mémoire propres au microprocesseur, connus sous le nom de registres de mémoires.

- *Convertisseur analogique numérique*

La conversion analogique numérique a pour rôle de transformer le signal analogique récupéré à partir des capteurs en données numériques manipulables par les unités restantes du microcontrôleur. Cela englobe deux étapes principales. La première étape se rapporte à l'échantillonnage du signal selon une certaine fréquence définie par la loi de **Nyquist**. La deuxième étape consiste à encoder la valeur échantillonnée en une série de bits regroupés sous forme d'octets.

- *Convertisseur numérique à analogique*

Cela correspond à la transformation d'une donnée numérique encodée sur une série de bits en une valeur analogique applicable aux actionneurs. Dans le contexte courant, cela correspond à la valeur appliquée pour la commande de la valve de refroidissement.

- *Unité MDI*

Cette unité intervient au niveau de la conversion numérique à analogique dans le cas de signaux ayant des amplitudes dépassant les capacités d'un simple convertisseur numérique à analogique. L'utilité d'une telle méthode se rapporte au fonctionnement de l'élément chauffant dont les besoins d'alimentation électrique restent relativement élevés. La méthode utilisée s'intitule **modulation de largeur d'impulsions**. Elle consiste en la génération d'une série d'impulsions numériques qui servent à moduler un signal de référence. Dans le cas courant, cela correspond au signal d'amplitude maximale pouvant être appliquée à l'élément chauffant. Par conséquent, le signal produit sera en moyenne égal à la valeur numérique convertie.

L'implémentation du système d'acquisition a eu recours au microcontrôleur **ADuC845** de la compagnie *ANALOG DEVICES*. Ce microcontrôleur intègre tous les éléments cités antérieurement. Les spécifications de ce microcontrôleur sont jointes à l'annexe B.

2.2. Communication matériel-logiciel

La présente section décrit le protocole de communication établi pour la communication bilatérale entre le PC et le système DAC. Il faut rappeler que les informations échangées se rapportent aux requêtes initiées par le logiciel de contrôle ayant pour objectif l'acquisition des données et l'application des commandes de contrôle. Elles se rapportent aussi aux réponses des requêtes d'acquisition de données renvoyées par le système matériel DAC. La figure 2.11 présente la structure du protocole de communication élaboré. Il prend la forme de deux structures d'informations qui seront introduites au niveau du chapitre 4. La première structure correspond à l'information envoyée du PC vers le système matériel DAC. La deuxième structure définit l'information renvoyée dans la direction opposée, soit du système DAC vers le PC.

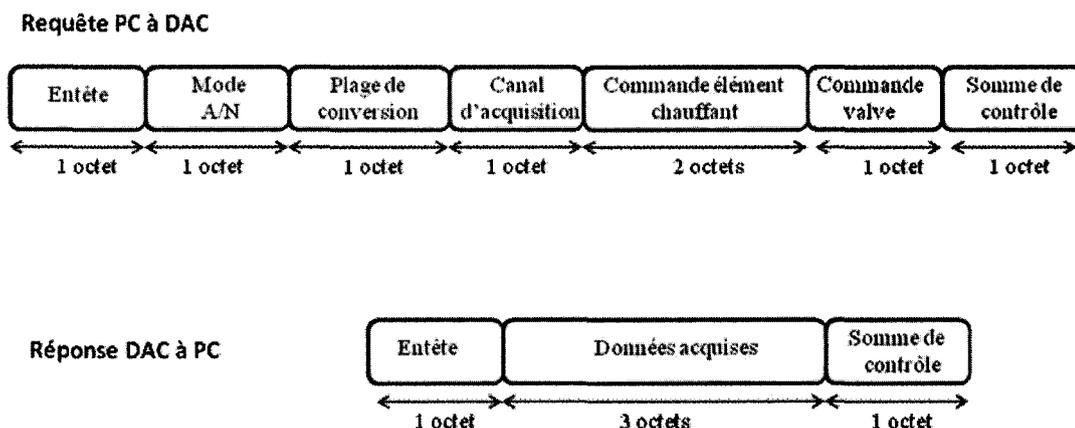


Figure 2.11 Structure du protocole de communication

2.2.1. Requête PC à DAC

L'unité de communication entre le PC et le système DAC se nomme télégramme. Ce dernier est défini sur 8 octets quant aux requêtes PC à DAC. Il comporte les éléments suivants :

- *En-tête*. L'en-tête marque le début d'un nouveau télégramme. Elle est assignée à la valeur fixe 85.
- *Mode A/N*. Il s'agit d'une configuration du convertisseur analogique/numérique qui permet de sélectionner le convertisseur à activer. Il est à mentionner en effet que le microcontrôleur utilisé comporte deux convertisseurs analogique/numérique indépendants. Cette même donnée sert aussi à fixer le mode de fonctionnement du convertisseur. La valeur utilisée pour la configuration du système DAC courant se rapporte au type de conversion par initiation. Ce dernier impose l'activation effective et périodique du processus de conversion d'un signal analogique donné en sa valeur numérique équivalente.

- *Plage de conversion.* Cette donnée codée sur un octet est utilisée pour la configuration du convertisseur analogique numérique. Elle permet de spécifier la plage maximale dans laquelle le signal à convertir varie.
- *Canal d'acquisition.* Il s'agit de l'information de sélection du capteur à considérer au niveau de l'acquisition initiée par la requête courante. Cette information agit directement sur le composant multiplexeur ayant à ses entrées les signaux des capteurs supportés.
- *Commande de l'élément chauffant.* Cette valeur encodée sur deux octets réfère à la commande appliquée à l'élément chauffant du système DAC. La proportion de cette valeur par rapport à son maximum possible correspond à la proportion de la puissance à imposer relativement à la puissance maximale supportée par l'élément chauffant. Étant donné que le contrôle de ce dernier est réalisé suivant la méthode de la modulation de largeur d'impulsions, cette proportion s'identifie au rapport cyclique des impulsions de modulations.
- *Commande de la valve.* Il s'agit de la valeur de la commande à appliquer à la valve de refroidissement du système DAC. Similairement à l'encodage utilisé pour la commande de l'élément chauffant, la proportion de la donnée courante par rapport à la valeur maximale qu'elle peut prendre est équivalente au pourcentage d'ouverture de l'orifice d'évacuation de l'air.
- *Somme de contrôle.* Cette donnée sert à confirmer la validité des données transmises. Elle représente la somme des 7 octets décrits précédemment. La même somme est censée être recalculée lors de la réception du télégramme afin d'être comparée avec la somme transportée. Une divergence entre les deux sommes calculée et reçue confirme l'invalidité du télégramme. Le cas échéant, le télégramme est rejeté et un nouveau télégramme est renvoyé.

2.2.2. Réponse DAC à PC

La réponse envoyée par le microcontrôleur vers le PC encapsule la donnée relative à un capteur donné. Elle est définie sur une plage de 5 octets et se structure comme suit :

- *En-tête.* L'en-tête marque le début d'un nouveau télégramme. Elle est assignée à la valeur fixe 153.
- *Données acquises.* La valeur correspondante est encodée sur 3 octets. Il s'agit du résultat de la conversion analogique à numérique du signal extrait au moyen d'un capteur. Ce dernier est sélectionné au niveau du champ du canal de sélection de la requête dont le télégramme courant constitue la réponse. L'encodage en 3 octets est justifié par la précision du convertisseur analogique à numérique qui s'étale sur une plage identique. L'ordre d'apparition des octets est équivalent à l'ordre décroissant de leur signification au niveau de la donnée originale.
- *Somme de contrôle.* Cette donnée sert à confirmer la validité des données transmises. Elle représente la somme des 4 octets décrits précédemment. Le même processus de validation utilisé lors de l'analyse de la validité des requêtes est repris aussi au niveau de la validation des réponses.

2.2.3. Synchronisation de la communication

Le protocole décrit implémente un mode communication semi-synchrone. En effet, l'envoi d'une requête d'acquisition par le PC vers le système matériel DAC entraîne la partie logicielle dans un cycle d'attente de la réponse correspondante. Ce cycle est achevé lors de la réception de la donnée requise ou lors de l'écoulement de la limite du temps d'attente. Cette limite est une constante interne de la partie logicielle du système. La figure 2.12 illustre les diverses situations de synchronisation de la communication

entre le PC et le système DAC. La synchronisation des communications entre les parties logicielles et matérielles constitue un facteur déterminant pour les performances globales du système. En effet, ceci permet de garder en permanence le système dans un état prévisible. Cela en définit amplement la robustesse et la fiabilité.

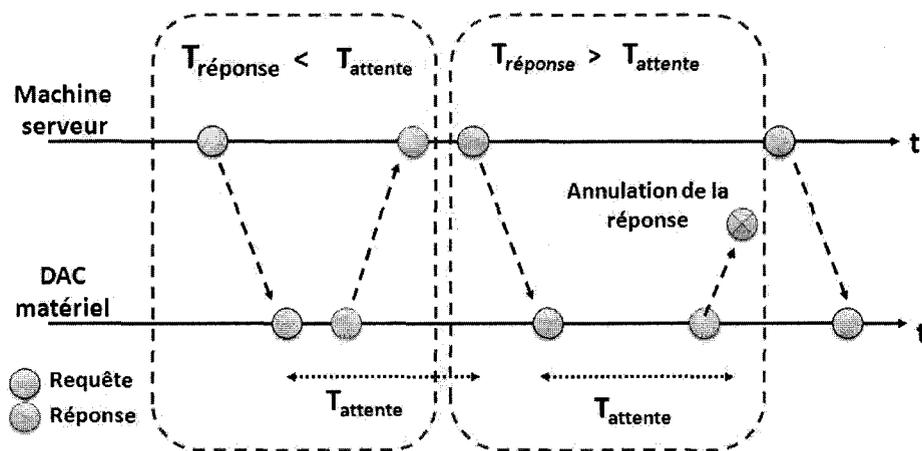


Figure 2.12 Synchronisation de la communication entre le PC et la partie matérielle du système DAC

CHAPITRE 3 : ANALYSE LOGICIELLE DU SYSTÈME D'ACQUISITION ET DE CONTRÔLE

Le présent chapitre a pour objectif de présenter les divers éléments relatifs à l'analyse de la partie logicielle du système d'acquisition et de contrôle (DAC). Ceci englobe les requis essentiels auxquels répond le logiciel conçu du point de vue fonctionnel et qualitatif. Les choix technologiques et architecturaux sur lesquels est fondée sur la conception de l'application seront aussi abordés.

3.1. Requis fonctionnels

3.1.1. Spécifications générales

Le système doit s'assurer de réaliser deux fonctions essentielles. L'acquisition de données en temps-réel est la première raison d'être du système global. Il s'agit d'initier périodiquement l'extraction des données descriptives du procédé étudié. Ces données seront récupérées à la suite de chaque requête d'extraction pour être transmises à d'autres unités d'exploitation de données. Un deuxième volet fonctionnel du système touche à la régulation du procédé industriel en cours. Effectivement, les données acquises sont exploitées à des fins de contrôle.

La figure 3.1 illustre les fonctionnalités globales du système logiciel conçu ainsi que les rapports qui existent entre eux. Le diagramme de cas d'utilisation qui s'y rapporte est donné à l'annexe A. Il faut mentionner que ce diagramme illustre de manière plus technique les fonctionnalités du système selon les normes d'analyse logicielle en vigueur.

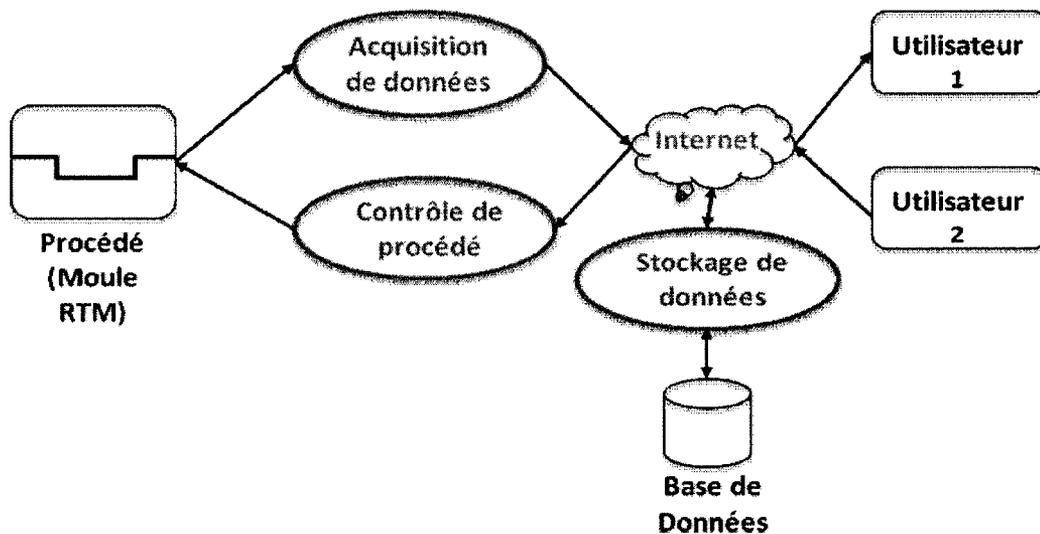


Figure 3.1 Diagramme des fonctionnalités globales du système logiciel d'acquisition et de contrôle

En plus des fonctionnalités principales d'acquisition et de contrôle, la figure 3.1 expose d'autres fonctionnalités auxiliaires. En effet, le système aura pour mandat de permettre la réalisation des fonctionnalités interactives suivantes :

- configuration des unités de contrôle à distance,
- définition des consignes de contrôle à distance,
- enregistrement permanent des données acquises et de la configuration du système de manière locale ou/et à distance.

Une explication plus détaillée de l'ensemble des fonctionnalités énoncées au cours de la section courante est donnée dans la suite de ce chapitre.

3.1.2. Requis d'acquisition de données et de contrôle

L'acquisition des données caractéristiques d'un procédé implique l'intervention d'un utilisateur humain. Ce dernier a pour rôle principal de configurer l'acquisition en ce qui a trait à la sélection des sources de données pertinentes à l'utilisation parmi les diverses sources de données disponibles. Cet utilisateur doit avoir le statut de super-utilisateur, qui confère à son détenteur les droits de configuration du système. Notons que plusieurs super-utilisateurs ne peuvent coexister simultanément pour éviter tout conflit d'accès et de configuration.

La fonctionnalité d'acquisition implique aussi l'intervention d'un module matériel, qui gère la panoplie de capteurs existants. Ce module matériel est en contact direct avec le procédé et sert d'intermédiaire entre ce dernier et le système logiciel d'acquisition et de contrôle.

Le système en cours encapsule aussi des fonctionnalités de contrôle de procédés. Ces dernières nécessitent une phase de configuration en termes de paramètres et de consignes de contrôle. Ces configurations sont effectuées de la part du super-utilisateur. Toutefois, le processus de régulation en soi n'a d'effet que sur le procédé lui-même, et ne fait intervenir aucun utilisateur. Cela est réalisé au moyen d'un module matériel, dont le rôle est de gérer divers actionneurs.

L'initiation du contrôle coïncide avec le démarrage de l'acquisition. Notons que cette fonctionnalité dépend étroitement de l'acquisition des données dans le sens où les données acquises servent de données d'entrée à l'unité de contrôle. Par conséquent, les actions du système relatives au contrôle du procédé succéderont à celles qui se rapportent à l'acquisition des données.

Étant donné l'aspect temps-réel auquel se plie la réalisation du système logiciel, les fonctionnalités d'acquisition et de contrôle s'effectuent de manière cyclique. Un tel

processus continue jusqu'à l'écoulement de toutes les consignes spécifiées ou jusqu'à l'arrêt intentionnel du système de la part de l'utilisateur aux privilèges élargis (super-utilisateur).

La figure 3.2 présente le diagramme de séquence des deux cas d'utilisation abordés précédemment. Ce dernier relate un scénario typique d'exécution des deux cas d'utilisation. Vu les dépendances qui existent entre ces fonctionnalités majeures du système, il s'avère judicieux de les traiter au sein d'un scénario unique.

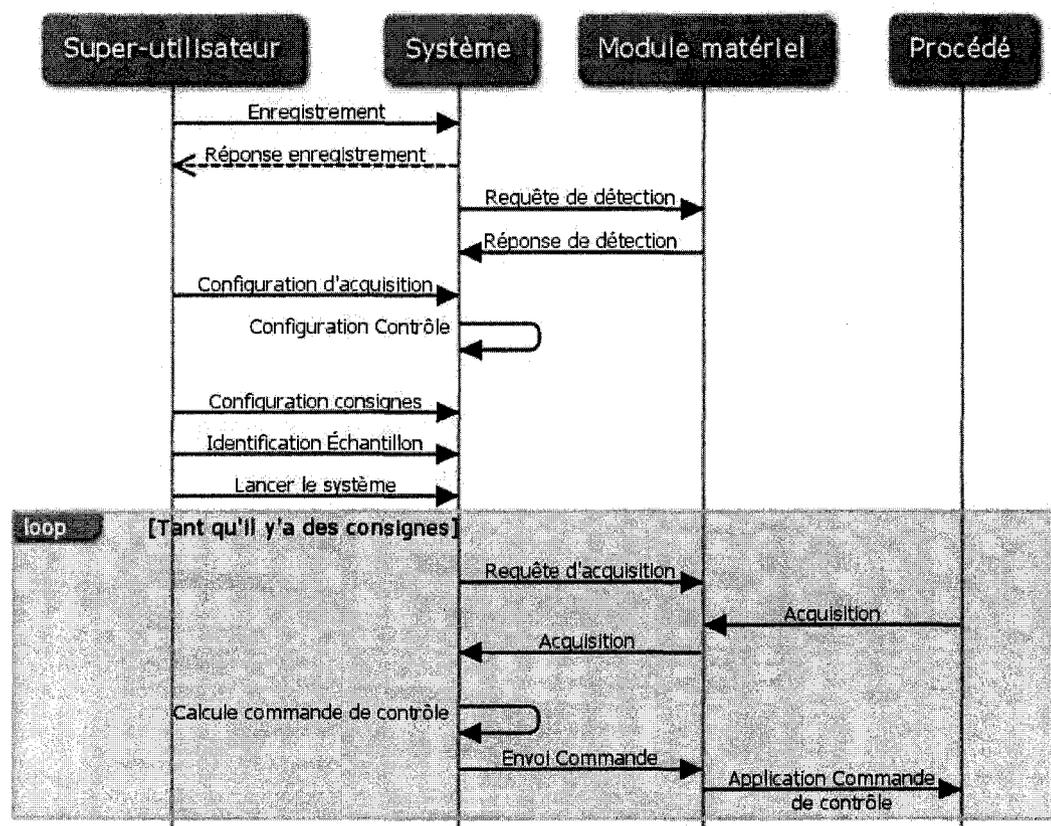


Figure 3.2 Diagramme de séquence relatif aux cas d'utilisation de l'acquisition de données et du contrôle

Le diagramme de la figure 3.2 repose sur le formalisme UML. Il présente sommairement les différentes étapes nécessaires pour la réalisation des cas d'utilisation relatifs à l'acquisition de données et du contrôle. Il existe quatre intervenants majeurs au niveau de l'exécution du scénario. Il s'agit évidemment du système en cours de réalisation, du super-utilisateur, du module matériel et du procédé. Notons l'absence d'interaction directe entre le système et le procédé. Néanmoins, l'inclusion de ce dernier élément dans le déroulement du scénario apparaît nécessaire, vu les relations qui existent entre cet élément et le système logiciel en développement.

3.1.3. Enregistrement des données et des configurations

Les fonctionnalités restantes supportées par le système touchent davantage à la gestion des données et des configurations. Elles offrent la possibilité d'un stockage permanent des données calculées par le système ou acquises à partir du procédé au cours d'une exécution complète du scénario présenté précédemment. Cet enregistrement peut se faire de manière locale sur la machine hôte de l'application logicielle qui gère les interactions avec l'utilisateur, ou à distance en faisant intervenir des supports de données accessibles à partir d'autres machines localisées à distance. Par conséquent, l'importation des données suivra les mêmes spécifications et pourra s'effectuer à partir de supports locaux ou distants. Notons qu'une exportation locale des données et des configurations s'effectue au niveau de fichiers de données formatés en conséquence. Par ailleurs, une exportation à distance des données fera intervenir un serveur de base de données.

Les figures 3.3 et 3.4 illustrent les diagrammes de séquence relatifs aux fonctionnalités d'enregistrement et d'importation des données respectivement. Notons que les diagrammes font abstraction de la nature du support d'enregistrement ou d'importation de données.

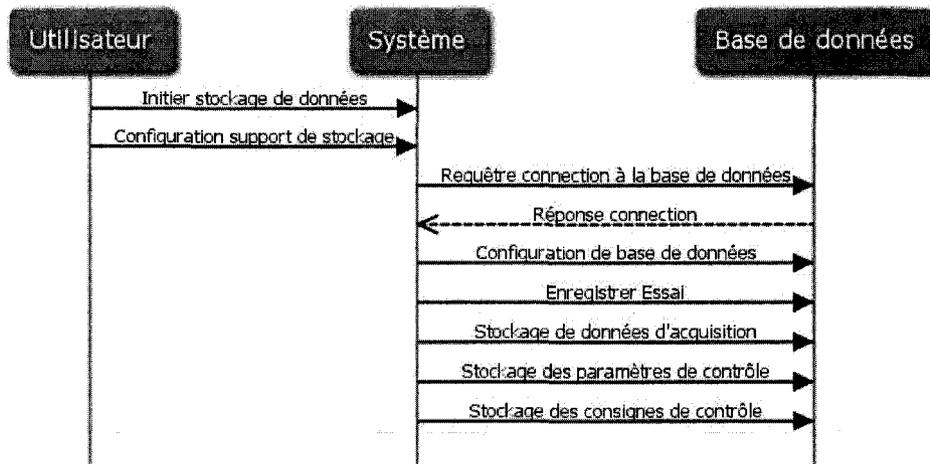


Figure 3.3 Diagramme de séquence pour l'exportation de données

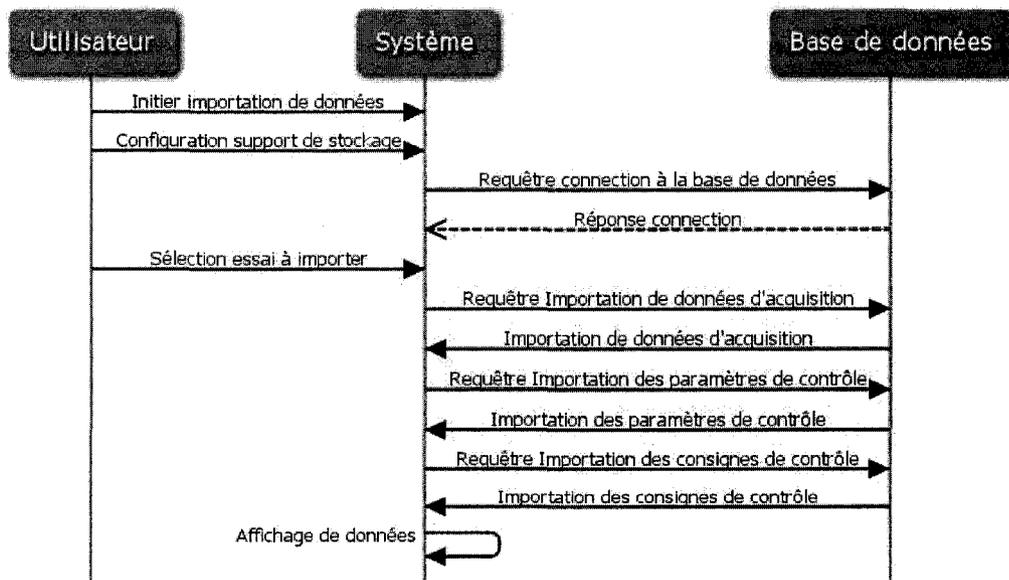


Figure 3.4 Diagramme de séquence pour l'importation de données

En plus de la nature distante des fonctionnalités de stockage et d'importation des données, l'ensemble des interactions des utilisateurs avec le système peut aussi s'effectuer à distance. En effet, il n'est pas nécessaire que l'interface offerte à l'utilisateur soit en contact physique direct avec le module matériel responsable de la gestion du procédé, à travers une machine hôte commune. Une distribution des différents éléments qui rentrent en interaction demeure possible à travers les réseaux informatiques. Étant une opération critique, il est préférable que le contrôle du procédé soit en contact direct avec le module matériel responsable de gérer ledit procédé. Ceci assurera une fiabilité accrue des algorithmes de contrôle, et permettra de mieux garantir les performances du système de contrôle.

3.2. Requis qualitatifs

Le système doit se plier à plusieurs spécifications qualitatives, imposant diverses contraintes au niveau qualitatif. Ces requis touchent entre autres à la robustesse du système, à son niveau d'interactivité et à la transparence par rapport aux détails relatifs aux dépendances architecturales. Par ailleurs, sa capacité à évoluer et son entretien demeurent des facteurs d'évaluation importants.

3.2.1. Robustesse

Ce requis se rapporte à la capacité du système à avoir un fonctionnement normal dans le cas de circonstances d'utilisation atypiques. Ces circonstances pourraient correspondre à une manipulation incorrecte de la part de l'utilisateur dans les cas suivants :

- entrées de données de format ou de type inapproprié (insérer une lettre dans un champ censé recevoir un nombre) ;

- activation de fonctionnalité dans des circonstances inappropriées (lancement de l'acquisition avant la configuration du système).

Les circonstances d'utilisation atypiques peuvent aussi se rapporter à des événements extérieurs inattendus. La probabilité d'occurrence de tels événements est accrue dans le cas de systèmes impliquant plusieurs composantes physiques en interaction. Ces événements de dysfonctionnement potentiel sont, entre autres, des pannes de transmission de données dans les réseaux informatiques ou de tout autre support utilisé, la désynchronisation des modules physiques en communication ou l'arrêt exceptionnel d'une composante du système. Il est important que le système détecte l'occurrence de tels événements et réagisse de manière prévisible.

3.2.2. Interactivité

Cette spécification est une mesure qualitative de la facilité de manipulation du système de la part des utilisateurs potentiels. Elle dépend étroitement de l'organisation de l'interface graphique du système et de son niveau de conformabilité avec les normes de conception des interfaces graphiques en vigueur. Par ailleurs, le niveau de transparence que possède l'interface graphique par rapport aux détails de conception n'ayant pas rapport avec les objectifs métiers du système, s'avère aussi être un élément important qu'il est nécessaire de considérer dans l'évaluation du niveau d'interactivité. Dans le cas courant, il s'agit davantage des détails de configuration du module matériel et du support de la communication en réseau. Ainsi, le système en cours doit avoir un niveau poussé d'interactivité de sorte à offrir une interface très intuitive aux utilisateurs ayant des connaissances dans le domaine de l'acquisition et du contrôle. Aussi, est-il important que l'utilisation du système ne nécessite aucune connaissance préalable que ce soit en matière de communication répartie dans le réseau, ou de conception matérielle.

3.2.3. Évolutivité et maintenabilité

Cette qualité mesure la capacité du système à intégrer de nouvelles fonctionnalités et la facilité avec laquelle cette intégration peut s'effectuer. Pratiquement, elle se mesure par rapport aux modifications nécessaires à l'ajout de nouvelles fonctionnalités et à leur impact sur les fonctionnalités déjà existantes. Cela dit, le système logiciel doit pouvoir évoluer afin de supporter de nouveaux modules matériels destinés à la manipulation de procédés industriels variés. Aussi, le support des différentes plateformes de communication devrait être aussi accessible avec un minimum de modifications. Il est à mentionner que la capacité d'évolution du système est en relation étroite avec son niveau de modularité. En effet, une décomposition accrue, mais non excessive des fonctionnalités du système en sous-modules accroît largement cette qualité d'évolution.

La maintenabilité d'un système logiciel est le niveau de prédisposition de ses fonctionnalités existantes à être modifiées, pour répondre à des mises à jour des requis fonctionnels implémentés ou à la résolution de problèmes de fonctionnement existants. Cette spécification non-fonctionnelle est à prendre en compte lors de la conception globale du système afin d'assurer la possibilité de réutilisation des fonctionnalités. La maintenabilité d'un système logiciel est étroitement liée à son niveau de modularité à l'instar de sa capacité d'évolution. Ces deux requis se traduisent généralement par des impacts communs au niveau de la conception du système.

CHAPITRE 4 : CONCEPTION LOGICIELLE DU SYSTÈME D'ACQUISITION ET DE CONTRÔLE

4.1. Architecture globale

Les choix architecturaux appliqués lors de la phase de conception du logiciel dépendent des spécifications fonctionnelles énoncées auparavant. Ainsi, les interactions à distance entre l'utilisateur et le système imposent le recours à une architecture répartie sur réseau. Par ailleurs, un serveur de base de données est nécessaire pour l'exécution des fonctionnalités d'exportation et d'importation de données. La figure 4.1 présente l'architecture globale du système DAC. Celle-ci expose les éléments physiques qui interviennent pour la réalisation des cas d'utilisation du système.

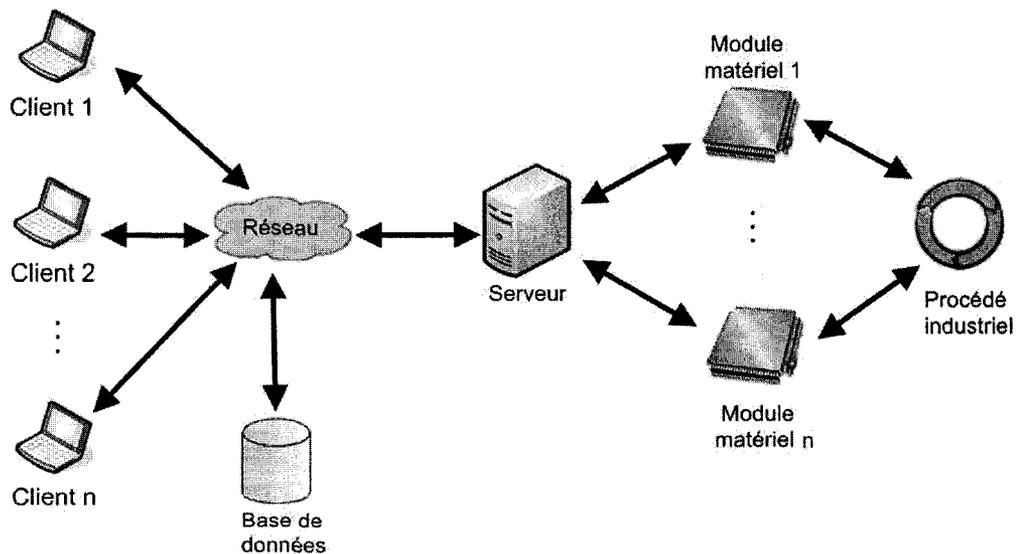


Figure 4.1 Architecture globale du système d'acquisition et de contrôle

Les nœuds clients de l'architecture hébergent les interfaces utilisateurs du système logiciel conçu. Celles-ci permettent aux utilisateurs d'exploiter diverses fonctionnalités, qui sont implémentées au sein d'un serveur localisé à distance. L'acquisition des données caractéristiques du procédé et sa régulation sont réalisées au moyen d'un module matériel dont le contrôle est géré par le même nœud serveur. Par ailleurs, une base de données offre la possibilité d'un enregistrement permanent des données. Cet enregistrement est essentiellement pris en charge par le nœud serveur. Toutefois, cette base de données reste accessible aux nœuds clients pour l'importation de données préalablement enregistrées. La figure 4.2 présente le diagramme de déploiement du système DAC, qui comprend les modules logiciels hébergés par les nœuds de l'architecture décrite à la figure 4.1.

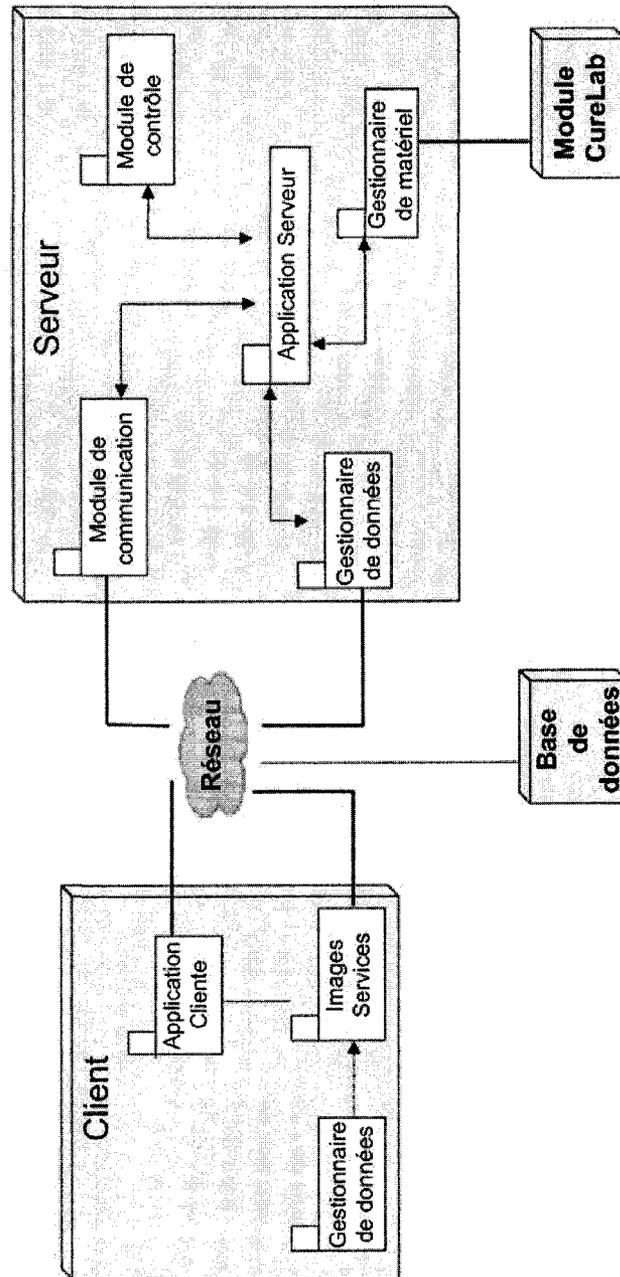


Figure 4.2 Diagramme de déploiement du système d'acquisition et de contrôle

La partie logicielle du système d'acquisition et de contrôle est composée de deux applications logicielles physiquement séparées, soit les applications cliente et serveur. L'application cliente constitue l'interface d'interaction de l'utilisateur avec le système. Elle regroupe principalement des éléments d'interface graphique et de support réseau pour l'accès à distance aux fonctionnalités de configuration, d'acquisition et de gestion des données. Par ailleurs, l'application serveur encapsule plusieurs modules correspondant respectivement aux requis fonctionnels cités précédemment. Ces modules s'énoncent comme suit :

- **Module de communication réseau.** Ce module regroupe l'ensemble des instances logicielles nécessaires à la communication du serveur avec les applications clientes. En d'autres termes, il s'agit de l'interface principale d'accès de l'application cliente aux fonctionnalités métiers du système, en l'occurrence la configuration de ce dernier ainsi que l'acquisition des données.
- **Gestionnaire de matériel.** Ce module englobe les fonctionnalités de communication du serveur avec les modules matériels. Il s'agit essentiellement des implémentations de protocoles de communication supportés par les modules matériels utilisés. Notons la forte dépendance de ce module avec le matériel dans le sens où il nécessite des mises à niveau substantielles dans le cas d'intégration de modules matériels additionnels.
- **Module de contrôle.** Il s'agit d'une implémentation des algorithmes de régulation de procédés. Le contrôle du procédé comprend plusieurs entités logicielles offrant des approches de régulation différentes. Ainsi, il est possible de varier la procédure de contrôle en fonction de la nature du procédé étudié.
- **Gestionnaire de base de données.** Ce module comprend l'ensemble des éléments logiciels nécessaires au stockage des données ainsi qu'à leur importation ultérieure. Certaines couches sous-jacentes de ce module possèdent une forte dépendance par

rapport aux supports de stockage potentiellement utilisés à partir d'interfaces d'accès hétérogènes.

4.2. Processus de localisation

Le caractère réparti du système logiciel d'acquisition et de contrôle implique un certain nombre de considérations additionnelles par rapport à la gestion des communications sur les réseaux informatiques. Ces considérations se rapportent aux approches de localisations mutuelles des différents nœuds qui forment le système réparti avant toute exécution des fonctions potentielles du système. D'ailleurs, les requis d'interactivité et de transparence imposent une prise en charge automatique du processus de localisation dans le sens où l'utilisateur ne doit pas être amené à effectuer de configuration relative à la localisation en réseau. Aussi, est-il impératif que le processus de localisation s'adapte, sans l'intervention de l'utilisateur, au changement des emplacements des nœuds du système.

L'architecture proposée prend en compte l'ensemble des contraintes énoncées précédemment. Effectivement, elle permet à une application cliente qui requiert les services du système, de détecter automatiquement l'emplacement de l'application serveur. Le processus de localisation proposé repose partiellement sur le principe des systèmes DNS. Ces derniers consistent en des serveurs dont le mandat est d'associer les adresses web avec les adresses IP des machines qui les hébergent. De manière similaire, le recours à un mécanisme d'identification pour associer le nom de l'application serveur avec son adresse IP dans le réseau s'avère être une approche de localisation pertinente. Cela est effectué par le biais d'un serveur de base de données accessible en permanence sur le réseau.

Deux étapes sont nécessaires pour la réalisation du processus de localisation. La première consiste en l'enregistrement de l'application serveur du système auprès du

serveur de base de données dédié aux opérations d'identification. Par la suite, l'application cliente consulte la liste des applications serveurs enregistrées et récupère les informations relatives à une application serveur donnée pour établir une connexion directe avec cette dernière. La figure 4.3 illustre un cas de figure du processus de localisation proposé.

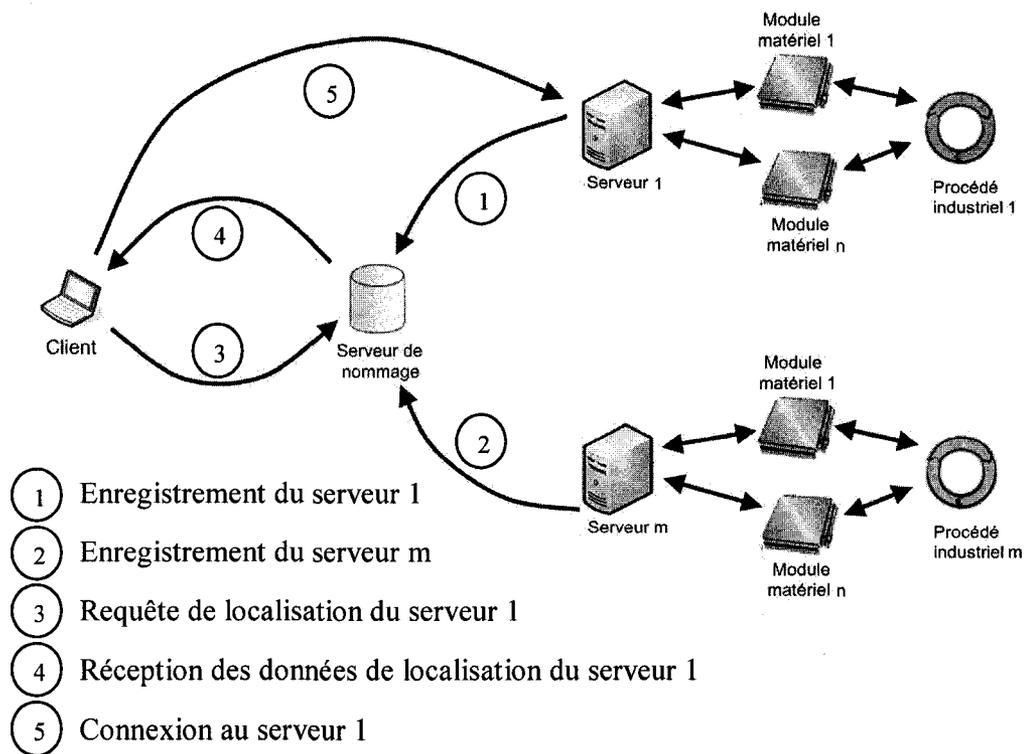


Figure 4.3 Mécanisme de localisation du système logiciel d'acquisition et de contrôle

4.3. Interface utilisateur

L'interface utilisateur du système DAC fait référence à l'application qui fait l'objet de cette étude, soit la caractérisation thermique des matériaux composites. Cela dit, le système permet d'effectuer l'acquisition de données thermiques comme la température et le flux thermique. Elle présente un aspect entièrement graphique offrant des points d'entrée de données et de sortie de résultats selon une organisation intuitive et conviviale. Cette interface comprend une série de fonctionnalités pour enregistrer un utilisateur, configurer le matériel, contrôler et configurer le procédé.

4.3.1. Enregistrement de l'utilisateur

L'étape d'enregistrement a pour but l'identification de l'application cliente. Cela s'effectue à travers une chaîne de caractères unique choisie par l'utilisateur. Cette identification est associée à l'emplacement logique de l'application cliente au sein du réseau informatique auquel elle est connectée. Cette position logique dans le réseau n'est autre que l'adresse IP de l'interface réseau active de la machine hôte de l'application cliente. La figure 4.4 illustre la fenêtre de dialogue qui est dédiée à cet effet.

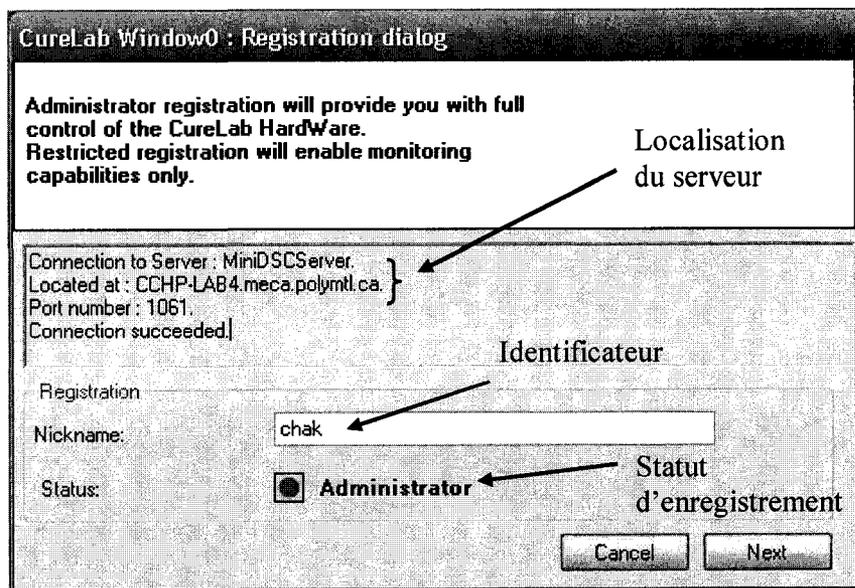


Figure 4.4 Fenêtre de dialogue pour l'enregistrement de l'utilisateur

Le processus d'enregistrement du client prend en charge automatiquement la détection de l'emplacement logique de l'application serveur à travers le processus de localisation énoncé antérieurement. L'attribution du statut de connexion au client en cours d'enregistrement est une fonctionnalité interne du système. Deux statuts de connexion sont possibles. Le statut d'administrateur (super-utilisateur) donne droit aux configurations du système et à son contrôle séquentiel, soit les opérations de lancement et d'arrêt. Le statut restreint limite les fonctionnalités du système aux activités de suivi et de gestion de données. La gestion des statuts de connexion s'effectue en conformité avec le requis d'unicité de l'utilisateur possédant tous les droits offerts par le système. Effectivement, le statut d'administrateur est attribué automatiquement à la première application cliente qui tente une connexion au système. Toute connexion subséquente se fera attribuer de facto un statut limité. Notons que les interactions décrites dans la suite concernent le statut administrateur.

4.3.2. Configuration des modules matériel

L'étape courante permet de sélectionner la liste des éléments matériels à prendre en compte lors du processus d'acquisition et de contrôle. Il s'agit dans ce cas-ci de capteurs de température et de flux thermique en provenance de deux types de modules matériels supportés. La figure 4.5 expose la fenêtre de dialogue utilisée.

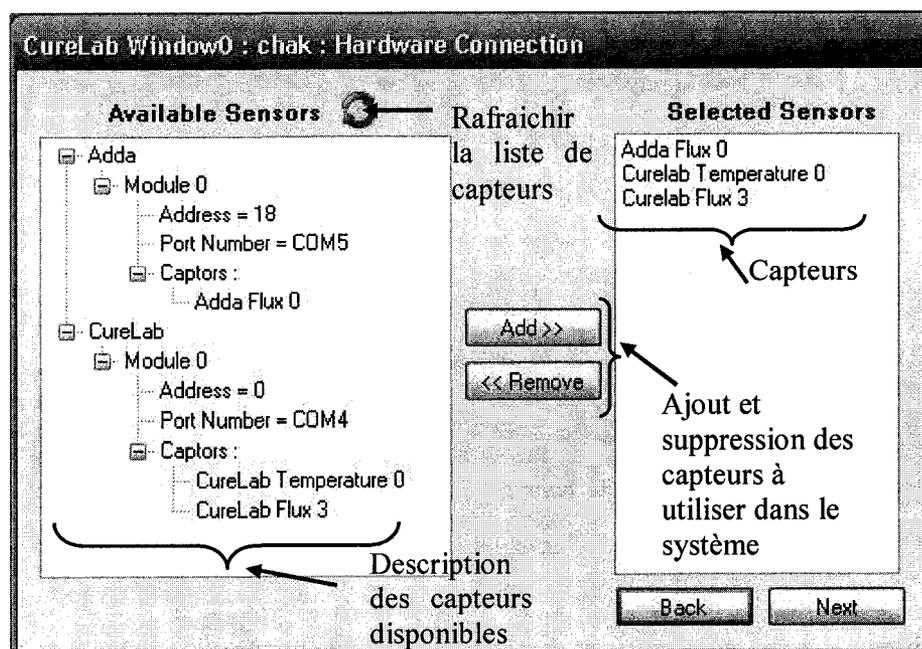


Figure 4.5 Fenêtre de dialogue pour la configuration matérielle

La détection des modules matériels actifs est aussi une fonction interne du système. Ceci s'effectue en parcourant tous les ports disponibles sur le canal de transmission considéré pour la communication entre l'application serveur et le module matériel. Dans l'application actuelle, il s'agit du port de transmission *RS232*. Pour chaque port disponible, un algorithme de détection est appliqué. Cet algorithme dépend généralement du matériel supporté. Toutefois, il reste possible d'utiliser un algorithme général, applicable à toutes sortes de modules matériels. Cela consiste en une tentative d'acquisition de données dont le résultat indique la présence du module matériel testé. Le critère de validité suppose la connaissance préalable des conditions dans lesquels opère le système. Concrètement, dans le cas de figure courant, les capteurs de température sont censés fournir une température équivalente approximativement à la température de la pièce où ils sont placés. Par ailleurs, les conditions optimales au moment du lancement d'un essai donné supposent l'absence de tout flux thermique. Par conséquent, les capteurs de flux devront retourner des valeurs quasi nulles.

4.3.3. Configuration du contrôle du procédé

L'interaction courante se rapporte à la configuration de l'algorithme de contrôle pour l'ensemble des contrôleurs supportés, et plus spécifiquement à la configuration des paramètres relatifs à l'algorithme choisi. La figure 4.6 montre la fenêtre de dialogue destinée à cette fin. Les algorithmes de contrôle supportés et leurs paramètres de fonctionnement sont détectables automatiquement. Cela offre l'opportunité d'avoir une liste évolutive d'algorithmes sans aucun remaniement du code source du système en cours de développement.

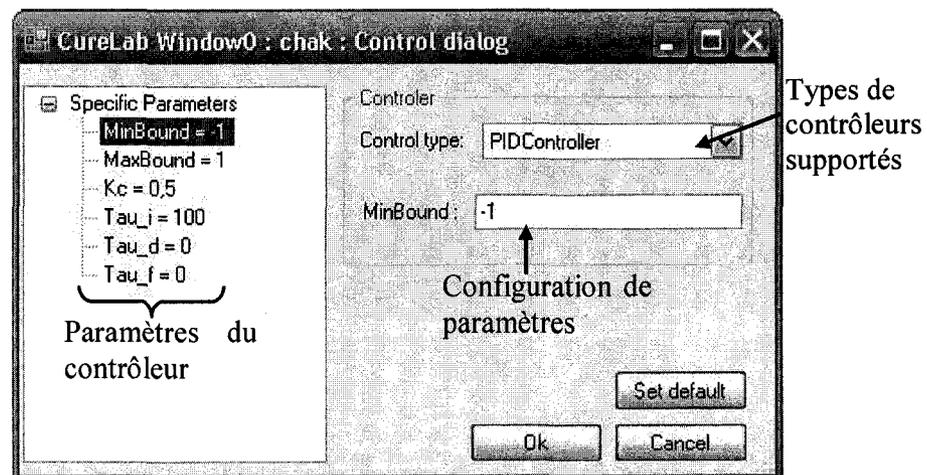


Figure 4.6 Fenêtre de dialogue pour la configuration du contrôle

4.3.4. Configuration des consignes de contrôle

La fenêtre de dialogue correspondant à cette étape permet de configurer les consignes de contrôle. L'application actuelle du système supporte deux types de consignes potentielles. En effet, il est possible d'appliquer une consigne fixe qui aboutit à une stabilisation de la variable contrôlée. La valeur de la consigne ainsi que sa durée sont des paramètres configurables. Le système offre aussi la possibilité d'imposer une contrainte linéaire à la variable de contrôle. Le cas échéant, la vitesse de variation et le seuil d'arrêt représentent les paramètres à configurer. Notons qu'une combinaison en série des types de contraintes supportées est possible. La consigne de contrôle dans l'application actuelle est la température de l'échantillon.

Le système impose une consigne de départ commune à tout essai afin d'obtenir des conditions initiales fixes choisies par l'utilisateur. Ces conditions sont associées à une période de stabilisation au-delà de laquelle les conditions du système sont supposées

stationnaires. La figure 4.7 présente l'élément d'interface graphique qui gère la configuration des consignes de contrôle :

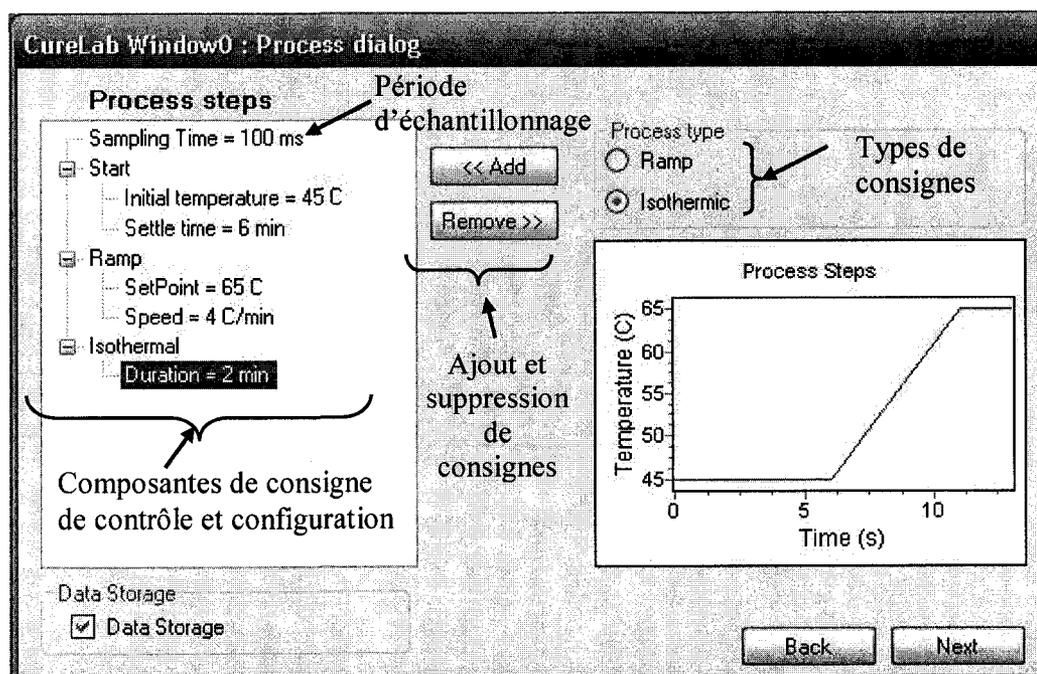


Figure 4.7 Fenêtre de dialogue pour la configuration des consignes de contrôle

Le même élément d'interface offre l'option de configurer la période d'échantillonnage pour l'acquisition des données caractéristiques du procédé. Ce paramètre correspond aussi à la période d'application des commandes de contrôle. Le choix de la période d'échantillonnage repose sur des critères qui se rapportent à la performance de l'acquisition et du contrôle du procédé.

4.3.5. Fenêtre principale

Les éléments d'interface graphique dédiés à la configuration du système sont principalement accessibles à partir de la fenêtre principale de l'application cliente. Cette dernière a pour objectif premier d'afficher en temps-réel les résultats du processus d'acquisition des données caractéristiques du procédé. Cela permet d'une part de faire le suivi de l'évolution des états de ce dernier, et d'autre part de valider les algorithmes de contrôle en vigueur. D'autres éléments auxiliaires figurent aussi au niveau de la fenêtre principale de l'application. Ils ont pour but de permettre une gestion plus efficace du fonctionnement global du système. En effet, un journal d'événements intégré à l'application permet d'énumérer l'ensemble des étapes par lesquelles passe le système. Cela est nécessaire pour déceler toute irrégularité par rapport au fonctionnement normal prévu. Enfin, comme l'application permet un partage d'informations auprès de plusieurs utilisateurs, il est utile que l'interface graphique du système logiciel permette une forme d'interaction entre les différentes applications clientes qui exploitent simultanément les fonctionnalités offertes. Cela se concrétise par l'ajout d'une zone d'échange de messages entre toutes les entités connectées. Cet aspect est ainsi renforcé par la liste des informations relatives aux utilisateurs connectés simultanément au système. Ces informations comprennent les identifications des usagers ainsi que leurs emplacements respectifs au sein du réseau informatique. La figure 4.8 présente la fenêtre principale du logiciel d'acquisition et de contrôle des données thermiques dans l'application courante.

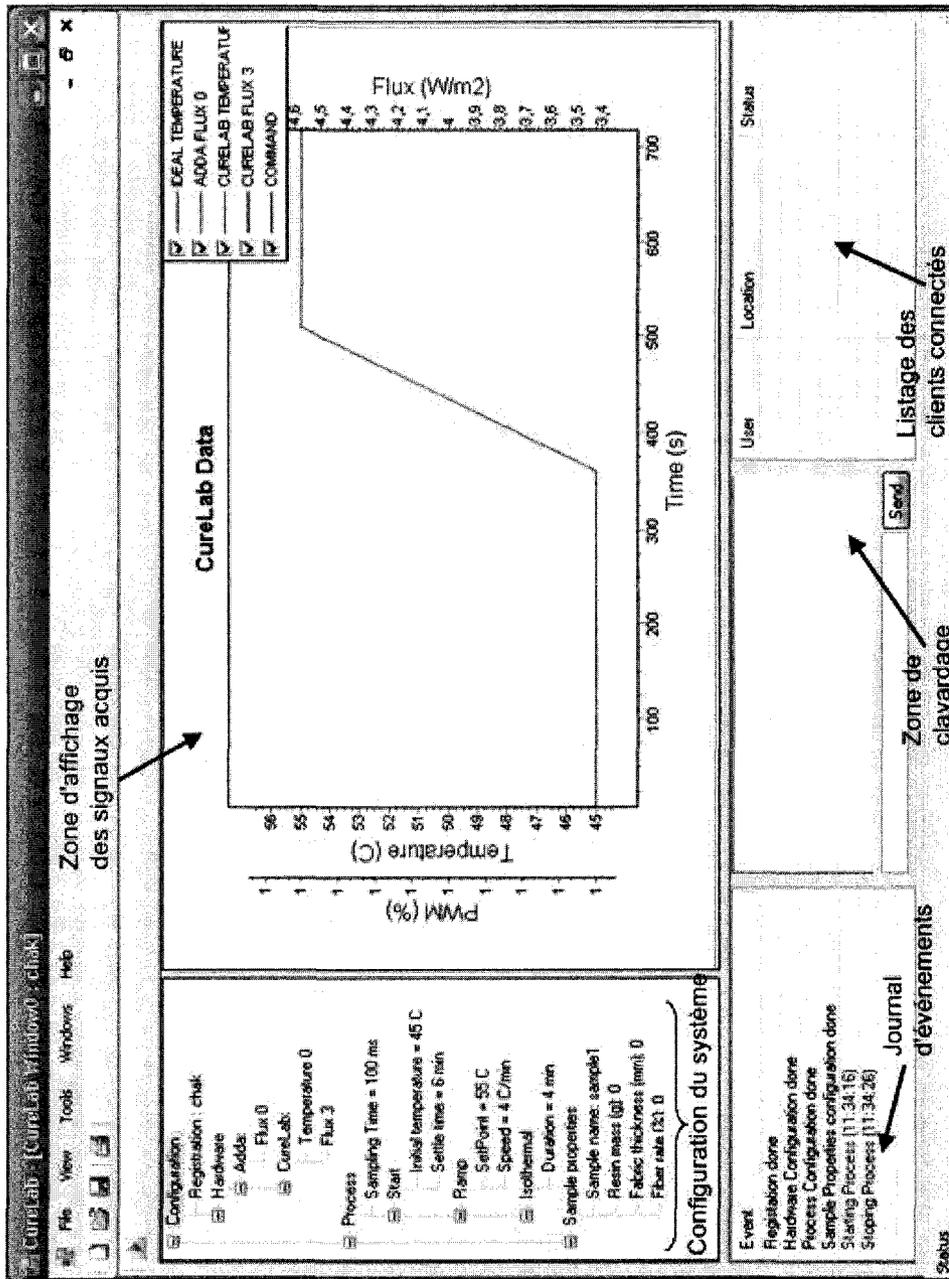


Figure 4.8 Fenêtre principale de l'interface graphique de l'application cliente du système DAC

Il paraît clair que le caractère compact de la fenêtre principale renforce l'interactivité dans le sens où l'accès aux diverses fonctionnalités s'effectue avec un nombre réduit d'opérations. Un point intéressant de l'application cliente réside dans la possibilité de lancer plusieurs fenêtres principales simultanément afin de gérer plusieurs procédés en même temps.

4.4. Architecture orientée services

Une caractéristique essentielle de l'architecture globale du système logiciel est sa répartition à travers le réseau informatique. En effet, l'accès à toutes les fonctionnalités est possible à distance tel qu'illustré précédemment. À cette fin, le support réseau du système repose sur les principes de la conception orientée-services. Cela consiste en l'encapsulation des fonctionnalités localisées à distance à travers les services hébergés par l'application serveur. Il s'agit en effet d'instances logicielles qui existent pendant tout le cycle de vie de l'application serveur et dont l'interface publique est accessible à distance. Cette interface n'est autre que l'ensemble des méthodes publiques encapsulées par l'instance logicielle elle-même. La figure 4.9 présente l'architecture orientée-services du système en cours.

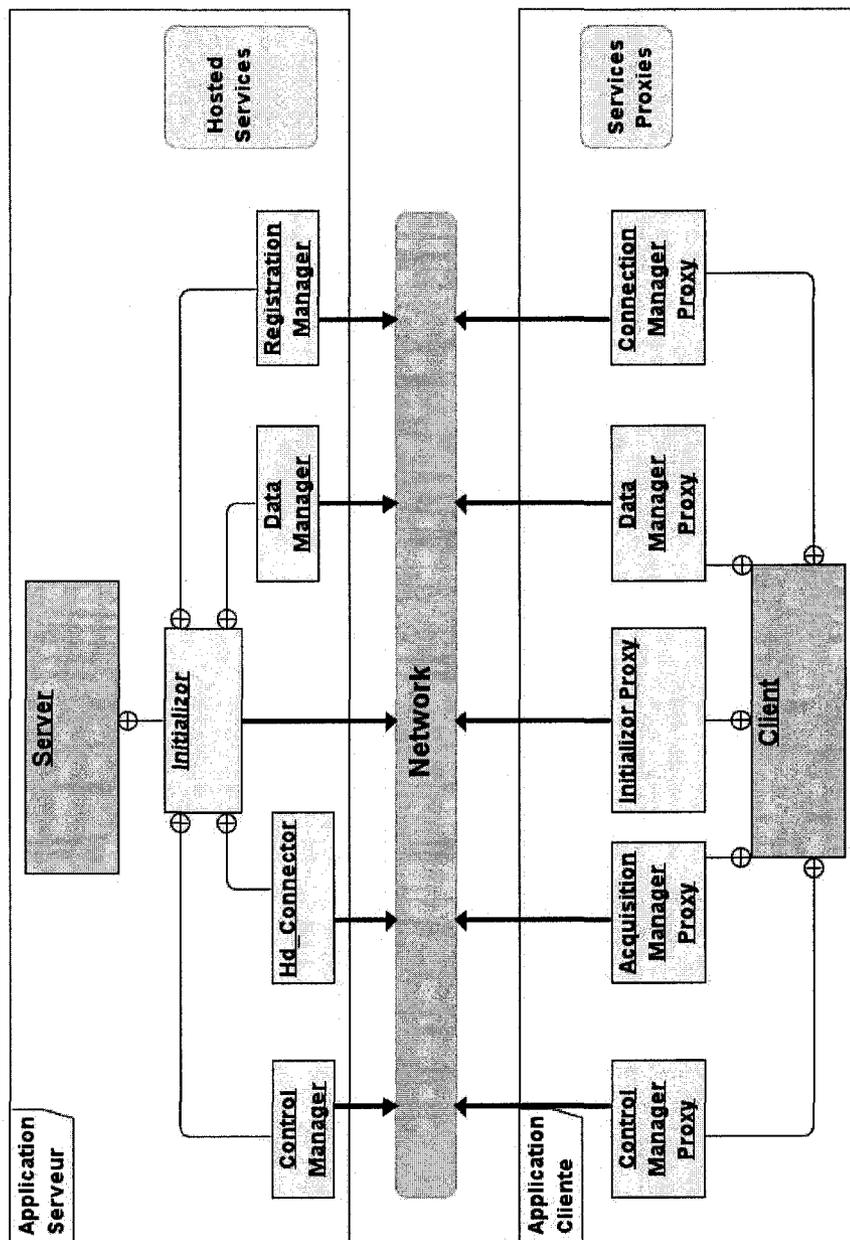


Figure 4.9 Architecture orientée-services du système DAC

L'architecture orientée-services du système présente quatre services majeurs correspondant chacun à une fonctionnalité métier de l'application. La création de ces services est prise en charge par un service supplémentaire dédié à cette tâche. Une telle instance est nécessaire pour gérer les situations de conflit lors de l'initialisation du système par le biais d'un mécanisme de gestion centralisée de l'instanciation des services métiers. Cette instanciation est initiée à distance par l'application cliente ayant les droits d'administration.

L'accès distant aux services hébergés au sein de l'application serveur est réalisé au moyen des instances *proxy* créées au sein de l'application cliente. Ces instances encapsulent respectivement l'image de l'interface publique de chaque service. Ainsi, l'exploitation des fonctionnalités accessibles à distance s'effectue par l'appel des méthodes des instances proxy locales.

Un tel choix architectural présente l'avantage d'encapsuler considérablement la gestion des transmissions de données à travers les réseaux, tel qu'expliqué au chapitre 1. Cela est rendu possible grâce aux plateformes logicielles sous-jacentes au système.

4.5. Environnement et plateforme

L'implémentation du système est réalisée majoritairement sur la base du standard **.Net 2.0**. En effet, les implémentations de ce dernier servent d'appui à la réalisation d'applications logicielles modulaires. Cette plateforme sous-jacente aux applications logicielles met à disposition du développeur plusieurs modules destinées à faciliter la réalisation d'interfaces graphiques conviviales ainsi que des applications aux architectures orientées-services fondées sur l'intergiciel **.Net Remoting** qui fait partie intégrante du même standard **.Net 2.0**.

L'implémentation du standard utilisée lors de la réalisation du système DAC est **Microsoft .Net 2.0**. Il s'agit d'une plateforme intégrée par défaut à la majorité des systèmes *Microsoft Windows* actuels. Les versions antérieures à la version Xp dudit système d'exploitation ne sont pas supportées. Par ailleurs, il existe diverses implémentations du même standard pouvant être intégrées à d'autres systèmes d'exploitation comme par exemple les différentes distributions de *Linux*. Cela offre l'opportunité d'une portabilité plus élargie pour le système DAC.

Notons que le module de gestion de contrôle illustré à la figure 4.2 est implémenté de manière entièrement indépendante au standard **.Net 2.0**. En effet, il s'agit d'un module externe réalisé dans le cadre d'un projet parallèle et dont l'implémentation en technologie native s'explique par son caractère fortement calculatoire, d'où le besoin d'une rapidité d'exécution logicielle avancée.

4.6. Détails de conception logicielle

Cette section développe les éléments de conception logicielle du système DAC. Il s'agit d'exposer en détail la structure des entités logicielles principales qui composent les applications cliente et serveur respectivement. Cette conception repose sur le paradigme orienté objet. En effet, ce dernier s'avère être une approche appropriée pour la conception du système dans la mesure où elle offre les moyens nécessaires pour répondre à toutes les spécifications énoncées antérieurement, notamment celles qui concernent les capacités d'évolution et d'entretien du système. Ces modules logiciels qu'englobent les applications cliente et serveur correspondent majoritairement aux paquetages conceptuels qui composent les structures logicielles des deux applications. Rappelons qu'un paquetage constitue un regroupement logique de sous-modules logiciels ayant un objectif fonctionnel commun.

4.6.1. Application Serveur

Le diagramme de paquetage correspondant à l'application serveur s'annonce selon la structure décrite à la figure 4.10. Dans l'analyse qui suit, l'emphase sera mise sur les paquetages noyaux de l'application. Il s'agit des paquetages de services, de leurs interfaces ainsi que du gestionnaire de matériel. Le module de contrôle a été produit dans le cadre d'un projet connexe et ne faisant objet que d'application au niveau du travail en cours. De ce fait, les détails de conception du module de contrôle ont été omis de cette étude. Par ailleurs, le paquetage de l'application serveur ne constitue que le point d'entrée de cette dernière. Il permet principalement de créer les entités appartenant aux autres paquetages à travers une seule et unique classe. Une analyse globale et sommaire des classes qui forment les paquetages à traiter permettra de mettre l'emphase sur le rôle que joue chacune des entités composantes ainsi que les types d'associations qui existent entre elles.

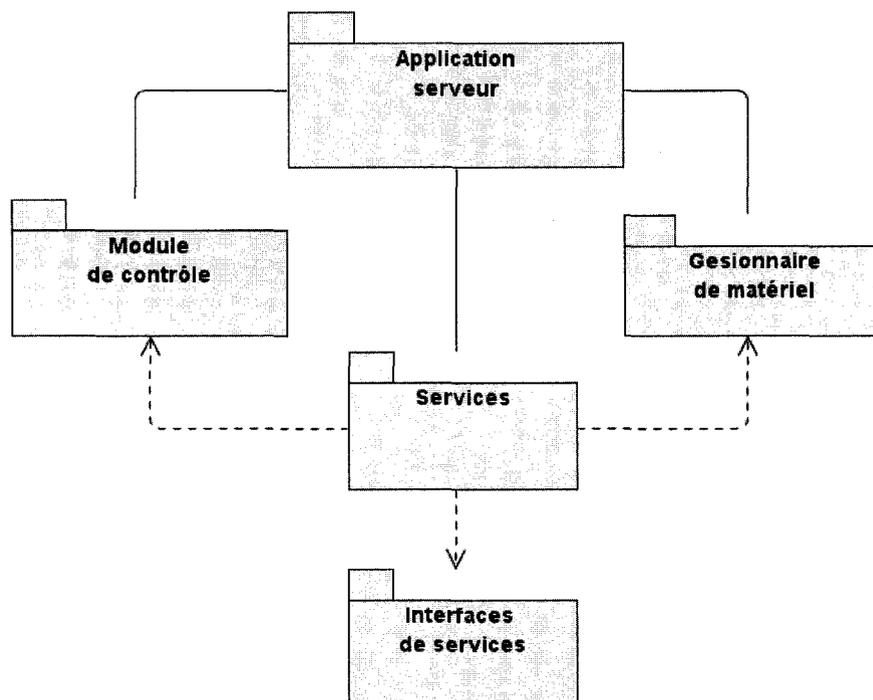


Figure 4.10 Diagramme de paquetage de l'application serveur

Gestionnaire de matériel

Ce paquetage contient l'ensemble des classes intervenant au niveau de la gestion des modules matériels pour l'acquisition et le contrôle. L'application à laquelle est destiné le système se rapporte à l'analyse de données thermiques et implique l'utilisation de deux modules matériels dédiés à cette tâche. Il s'agit des modules *ADDA* de la compagnie *Thermoflux*, et du module *CureLab*, un dispositif réalisé au sein du laboratoire de recherche qui supervise le projet courant. Les deux modules permettent d'effectuer l'acquisition des données de température et de flux thermique. La partie matérielle en

charge du contrôle est intégrée au module *CureLab*. La figure 4.11 illustre le diagramme de classe du paquetage du gestionnaire des modules matériels.

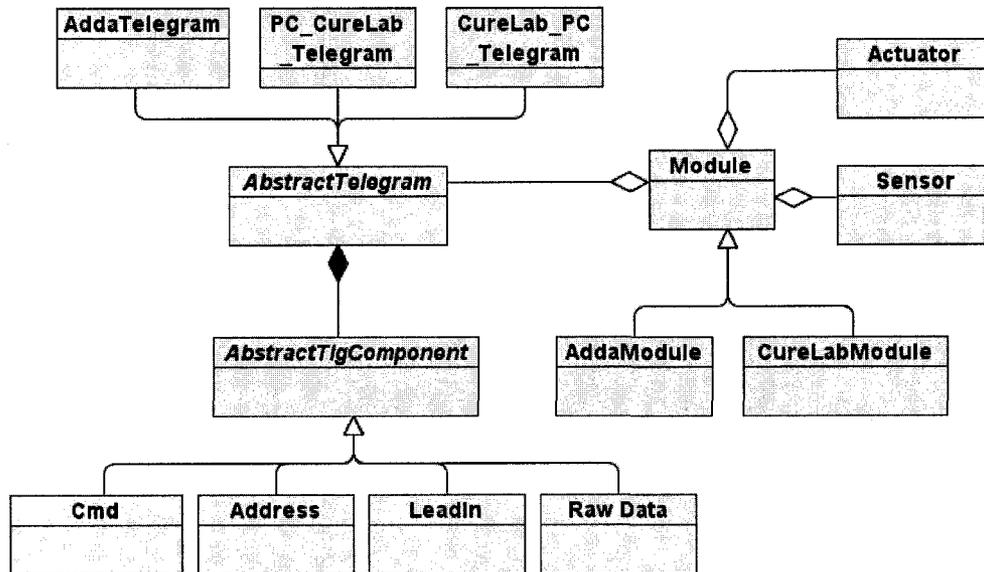


Figure 4.11 Diagramme de classe du paquetage du gestionnaire des modules matériels

La classe *Module* encapsule les propriétés relatives à un module matériel en ce qui a trait aux moyens de communication qu'il supporte, aux capteurs et aux actionneurs qu'il gère. Elle a pour rôle principale de prendre en charge les communications entre l'application serveur et un module matériel donné. Cette communication se fait à travers un protocole qui dépend étroitement du module matériel supporté.

L'implémentation du protocole de communication spécifique à un module matériel donné est réalisée au niveau d'une classe dérivée de la classe *Module*. Vu le support de deux types de modules matériels, deux classes dérivées correspondent chacune à un

module donné. Il s'agit des classes *AddaModule* et *CureLabModule*. Chaque module encapsule aussi une liste de capteurs et d'actionneurs. Ceux-ci sont représentés par les classes *Sensor* et *Actuator* respectivement. Celles-ci contiennent le type du capteur ou de l'actionneur et du module auquel ce dernier est connecté. L'entité du capteur prend aussi en charge le stockage de la donnée que ce dernier extrait du procédé étudié.

L'encapsulation de la structure du protocole de communication est réalisée à travers la classe abstraite *AbstractTelegram*. Cette dernière modélise la structure de base de l'unité d'information échangée lors d'une communication entre l'application serveur et le module matériel. Elle regroupe ainsi les éléments communs qui constituent cette unité d'information désignée par le concept de télégramme. Chaque élément du télégramme est représenté par la classe abstraite *AbstractTlgComponent*. Une telle entité constitue une généralisation à de nombreuses sous-composantes communes entre les télégrammes supportés. Ainsi sont encapsulées les entités d'en-tête et d'emplacements des nœuds communicants, à savoir d'une part la machine hôte de l'application serveur et d'autre part le module matériel. La commande imposée ou les données acquises à l'état brut figurent aussi parmi les sous-composantes du télégramme. Toutefois, la forme finale que prend le télégramme échangé dépend du module matériel auquel il correspond. Cela dit, il est de mise de dériver l'entité de télégramme en sous-classes spécifiques aux diverses situations de communications qui se présentent. Cela est réalisé au niveau des classes *AddaTelegram*, *PC_To_CureLab_Telegram* et *CureLab_To_PC_Telegram*.

Il paraît remarquable que l'organisation du paquetage de la gestion des modules matériels présente un aspect fortement modulaire qui donne lieu à des possibilités d'intégration de modules matériels additionnels autres que ceux supportés au niveau de la version courante. L'exercice d'une telle intégration a été effectué par la mise en œuvre d'une structure commune à deux modules matériels totalement hétérogènes, que ce soit au niveau de leur protocole de fonctionnement ou au niveau de leur organisation interne.

Services et interfaces

Ce paquetage regroupe l'ensemble des fonctionnalités métiers du système. Rappelons que ces fonctionnalités correspondent à l'acquisition des données caractéristiques du procédé étudié et à sa régulation. Les configurations du système en termes de contrôle et de support matériel constituent aussi une fonctionnalité dont l'intégration aux services accessibles à distance est de mise. Par ailleurs, d'autres fonctionnalités auxiliaires sont ajoutées au paquetage courant dans la mesure où elles sont indispensables au fonctionnement du système dans un environnement réparti et partagé potentiellement par plusieurs utilisateurs. Il s'agit des fonctionnalités d'enregistrement et de communication inter-clients.

À chaque service conçu correspond une interface donnée implémentée par ce dernier. Ces interfaces sont nécessaires pour la création des images clientes des services du côté des applications clientes. La figure 4.12 illustre le diagramme des services auquel a été annexé le diagramme de classes relatif aux interfaces. La classe *Initializor* a pour mandat principal de gérer les procédures d'instanciation des services identifiés. En effet, l'instance de cette classe correspond à un service statique créé au lancement de l'application serveur dans le but de gérer toutes les requêtes en provenance d'applications clientes.

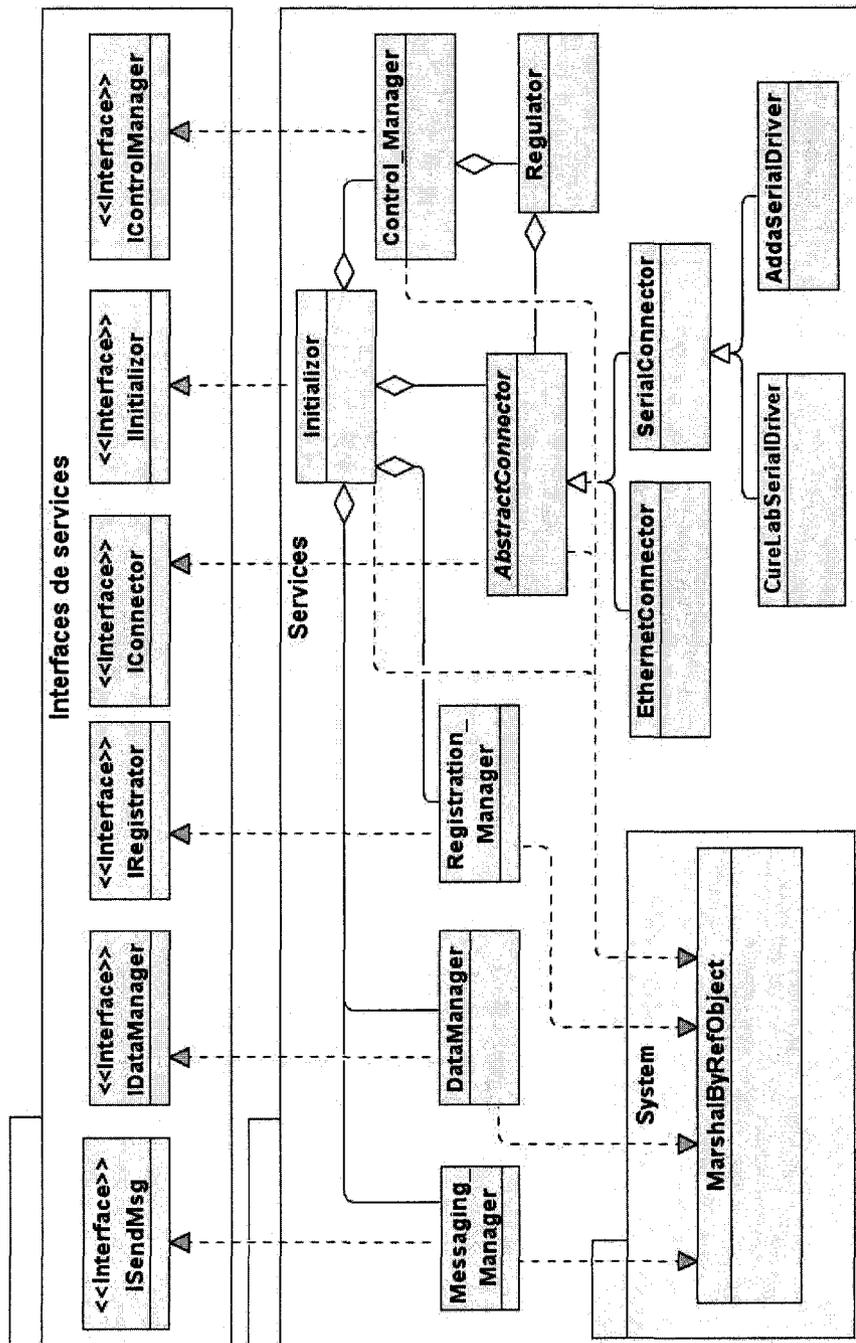


Figure 4.12 Diagramme de classe du paquetage de services et de leurs interfaces

Étant donné l'existence de deux modes de connexion conférant chacun des droits différents pour l'accès aux services et leur exploitation, il reste nécessaire de ne transférer les références des services à caractère restrictif qu'aux applications clientes qui y ont droit. Cela évite d'intégrer des requis de test de droits d'accès de manière éparpillée au sein des éléments de conception des services, d'où l'importance d'une gestion centralisée du processus d'obtention des références des services disponibles. Cela se concrétise au moyen du service *Initializer*.

La configuration du matériel, en termes de sélection des capteurs à prendre en compte lors du processus d'acquisition des données, s'effectue à travers la classe *AbstractConnector*. Cela a rapport direct avec la gestion des modules matériels, d'où l'inclusion de la liste de ces derniers au sein de la classe *AbstractConnector*. Notons que cette classe reste indépendante du canal de communication. En effet, cela est pris en charge au niveau de classes dérivées de la classe *AbstractConnector*. Celles-ci implémentent les aspects relatifs à la gestion des connexions en fonction du canal de transmission utilisé. Deux canaux sont introduits au niveau de la conception courante. Il s'agit des canaux de communication Série et Ethernet. Cela est réalisé au travers des classes *EthernetConnector* et *SerialConnector* respectivement. Toutefois, il est possible d'imaginer de nouveaux canaux de communication comme le canal Bluetooth par exemple.

Quoique l'exploitation des services de configuration matérielle se fasse de manière identique à tous les modules matériels supportés, à travers une interface distante commune, l'implémentation de cette dernière dépend étroitement des dispositifs matériels pris en charge. Ainsi se justifie la dérivation des classes *EthernetConnector* et *SerialConnector* en de nouvelles entités spécifiques à chaque dispositif supporté par le système. Les classes *CureLabSerialDriver* et *AddaSerialDriver* représentent ces entités et correspondent chacune aux modules *CureLab* et *Adda* respectivement.

L'initiation du processus d'acquisition de données fait appel à l'arborescence des classes décrites précédemment, qui constituent le chemin logique à emprunter pour atteindre les données acquises par les capteurs supportés. En effet, il faut rappeler que la même arborescence de classes encapsule les entités modélisatrices des modules matériels qui gardent des références aux capteurs des données pertinentes au processus d'acquisition. De ce fait, il paraît approprié d'inclure l'interface du service d'acquisition au sein de la classe *AbstractConnector*. Cette interface sert principalement à retourner l'état de la donnée encapsulée au niveau de l'objet capteur au moment de son appel.

La configuration des aspects du système correspondant au contrôle est prise en charge par la classe *ControlManager*. Celle-ci englobe la configuration des paramètres des algorithmes de contrôle ainsi que les consignes de contrôle. À cela s'ajoutent des fonctionnalités relatives au contrôle séquentiel du processus de régulation, en ce qui a trait au lancement et à l'arrêt de ce dernier. Cela dit, il paraît nécessaire d'inclure dans la classe courante, l'encapsulation de l'entité responsable du processus de régulation du procédé. Cette entité est réalisée à travers la classe *Regulator*. Effectivement, cette dernière implémente un algorithme cyclique effectuant successivement les opérations d'extraction de données du procédé et la régulation de ce dernier. Ainsi, se justifie la nécessité d'introduire des références aux instances de la classe *AbstractConnector* au niveau de l'entité courante. Les spécificités de l'algorithme de contrôle sont abordées au chapitre 5.

Les fonctionnalités de stockage des données et des configurations ainsi que leur importation sont implémentées au niveau de la classe *DataManager*. Cette dernière gère principalement plusieurs supports de stockage permanent de données, d'où son indépendance par rapport à ces derniers. La conception actuelle du logiciel prend en charge les supports *MySQL*, *Oracle*, *SQL Server* et *Access*. Toutefois, l'intégration de nouveaux supports reste accessible. Ces supports gèrent des bases de données de type

relationnel. Le format des tables d'enregistrement de données est unique à tous les supports de stockage.

La gestion des connexions des applications clientes au système est réalisée par la classe *RegistrationManager*. Ce service auxiliaire, mais pour le moins nécessaire, permet d'extraire les informations d'indentification des applications clientes. Cela est indispensable à la gestion des droits d'accès aux services et à la communication inter-clients. Cette dernière est réalisée au niveau de la classe *Messaging_Manager*, dont l'interface permet de transférer des messages à l'application serveur. Celle-ci aura pour responsabilité subséquente de retransmettre ces mêmes messages à toutes les applications clientes simultanément connectées au système.

Notons la dérivation commune de tous les services accessibles à distance de la classe *MarshalByRefObject* implémentée au sein du paquetage *System* de la plateforme .Net 2.0. Il s'agit d'une classe qui définit tous les aspects communs à un service distant donné.

4.6.2. Application cliente

L'application cliente est composée de trois paquetages majeurs. Il s'agit du paquetage regroupant les éléments de l'interface graphique de l'application cliente d'un côté, auquel s'ajoute le paquetage englobant les fonctionnalités de gestion locale des données à partir de supports de stockage permanent locaux. Ainsi, il reste possible de reprendre l'état et les résultats du système logiciel correspondant à une manipulation effectuée antérieurement dans le cas d'absence de connexion au serveur. Étant donné l'unicité des formats de stockage de données et leur indépendance des emplacements des bases de données, les mêmes fonctionnalités de gestion de ces dernières dans le cas d'accès à distance restent valables pour le cas d'accès local aussi. En d'autres termes, les fonctionnalités de stockage et d'importation locales des données sont réalisées à travers

un service hébergé localement. Par conséquent, le service de gestion de données de l'application serveur est repris au niveau de la conception de l'application cliente. Un deuxième service est aussi défini au niveau de l'application cliente. Il intervient au niveau de la communication inter-clients. Un troisième paquetage définit les interfaces d'accès à distance relatives aux deux services mentionnés plus haut. Cela donne lieu à la structure du diagramme de paquetage illustré à la figure 4.13 :

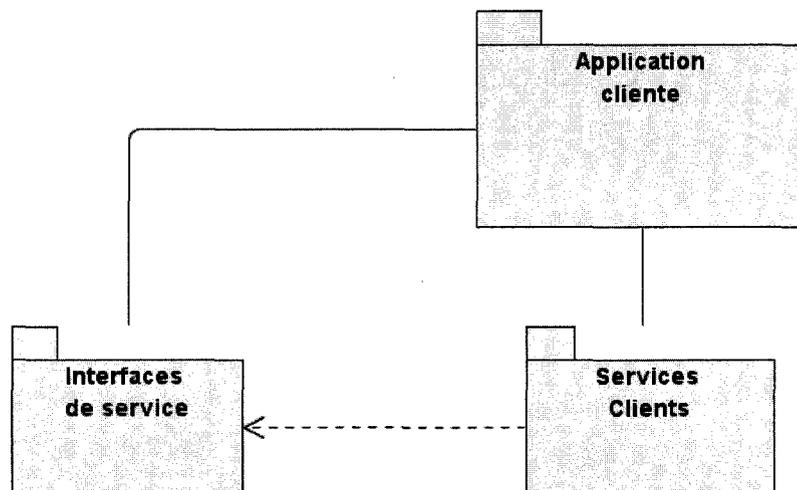


Figure 4.13 Diagramme de paquetage de l'application cliente

Le paquetage d'application cliente comporte l'entité d'entrée de l'application dont l'objectif est de créer l'ensemble des fenêtres graphiques de l'interface utilisateur. Ces éléments d'interface graphique sont tous dérivés de la classe *Form* définie dans le paquetage *System.Windows.Forms*. La figure 4.14 montre les classes que comporte l'application cliente.

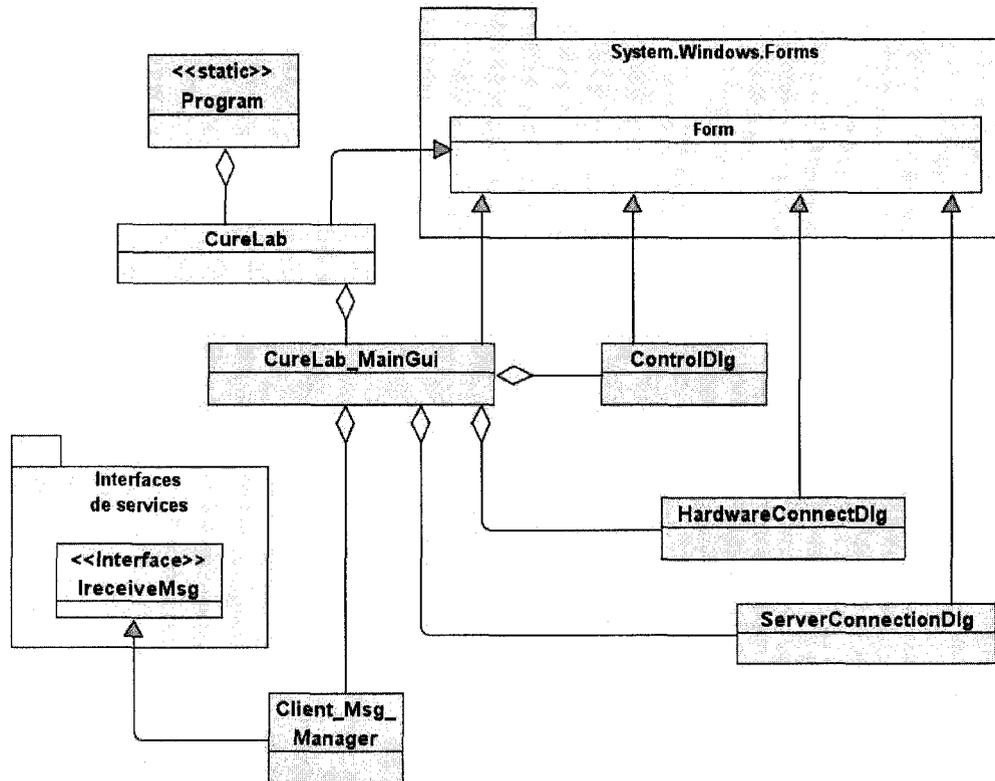


Figure 4.14 Diagramme de classe de l'application cliente et des éléments du paquetage des interfaces de services y étant relatifs

Notons que le diagramme ne présente que les éléments principaux de l'interface graphique de l'application dont l'introduction a été faite dans la section 4.3. Toutefois, il reste à mentionner l'existence d'autres éléments d'interface graphique définis au niveau du paquetage de l'application cliente et destinés à faciliter l'interactivité de l'application courante.

Une particularité que comporte l'application cliente consiste en son hébergement d'un service à distance dont l'exploitation est effectuée par l'application serveur. Ce service

modélisé par la classe *ClientMsgManager* intervient au niveau de la communication inter-cliente. Il permet d'instaurer un mécanisme de notification de l'envoi de messages en provenance d'autres applications clientes. En effet, chaque message envoyé par un client donné est redistribué aux autres clients connectés. La distribution est effectuée par l'application serveur à travers l'accès aux services hébergés respectivement au sein de chaque application cliente. Un tel algorithme évite ainsi d'avoir recours à un mécanisme de rafraîchissement périodique et consommateur de ressources en termes de temps CPU. Notons l'ajout d'une interface d'accès à distance à ce service au niveau du paquetage d'interfaces de services.

CHAPITRE 5 : ÉTUDE EXPÉRIMENTALE ET APPLICATIONS

Ce chapitre traite de l'application expérimentale du système d'acquisition et de contrôle décrit dans les chapitres précédents. Il s'agira de présenter l'algorithme de contrôle utilisé au niveau du système. La méthodologie d'optimisation des paramètres de contrôle est introduite aussi. Par la suite, la validation du fonctionnement global du système est effectuée par son application à la caractérisation thermique d'une résine thermodurcissable. Les résultats obtenus sont présentés et discutés.

5.1. Stratégie de contrôle thermique

Cette section décrit la stratégie de contrôle des données thermiques. Il s'agit de relater les spécificités de l'algorithme de contrôle choisi et d'en extraire les modalités de configuration et d'optimisation utilisées. Par la suite, une application du système d'acquisition et de contrôle à la caractérisation thermique d'une résine thermodurcissable est analysée.

5.1.1. Configuration de contrôle

Le contrôle d'un procédé consiste à maintenir un ou plusieurs états caractéristiques de ce dernier en conformité avec des consignes de contrôle préalablement établies. Dans le cas de l'application courante, il s'agit de réguler la température d'un système physique donné. À cette fin, le mécanisme de contrôle exploite les principes de la thermique. En effet, l'action de contrôle s'effectue sur le flux thermique traversant le système physique étudié. Cela est rendu possible grâce aux actionneurs de l'élément chauffant et de la valve de refroidissement.

La stratégie de régulation choisie pour le système DAC courant correspond au contrôle par rétroaction. Celle-ci présente plusieurs avantages de fiabilité et de performance incitant à son utilisation dans la majorité des cas de contrôle de complexité limitée. Cela constitue donc une stratégie fiable pour le système DAC thermique. La figure 5.1 illustre une boucle de commande par ordinateur adaptée au système DAC:

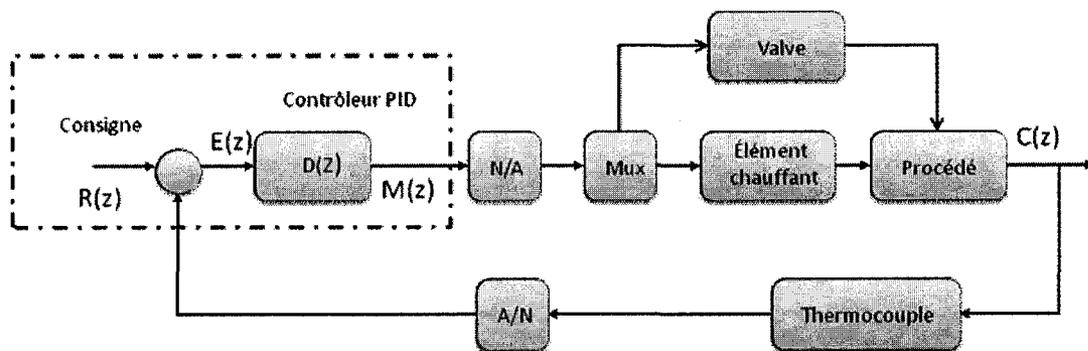


Figure 5.1 Boucle de commande par ordinateur adaptée au système DAC thermique

La régulation du procédé au niveau de la boucle de contrôle s'effectue à travers une composante dédiée implémentant un algorithme de contrôle donné. Un exemple pertinent de ce type d'algorithme est le contrôleur PID. Ce dernier présente plusieurs avantages pertinents quant à son intuitivité et sa rapidité d'implémentation. À cela s'ajoute aussi les résultats de performance intéressante auquel il donne accès dans le cas de système mono-variable. Rappelons que la fonction de transfert de ce type d'algorithme dans le domaine discret s'énonce comme suit :

$$D(z) = \frac{M(z)}{E(z)} = K_p - \frac{K_i T}{2} + \frac{K_i T}{1 - z^{-1}} + \frac{K_d}{T} (1 - z^{-1}) \quad (5.1)$$

où $M(z)$ constitue la commande à appliquer au système et $E(z)$ l'erreur de la variable contrôlée par rapport à la consigne de régulation. La configuration du montage sur lequel est appliqué le système de contrôle thermique est illustrée à la figure 5.2 :

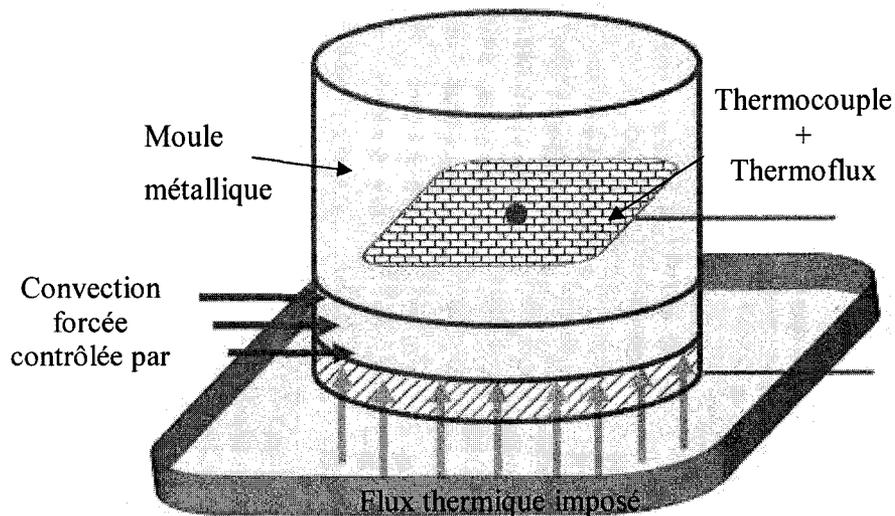


Figure 5.2 Schéma simplifié représentant la configuration du système physique sur lequel est appliqué le système DAC

L'objectif est de maintenir la température de l'emplacement où est situé le capteur thermocouple en adéquation avec les consignes de contrôle imposées. À cette fin, un élément chauffant est installé sur la base inférieure du cylindre. Le courant d'air généré à la sortie de la valve est appliqué latéralement au cylindre au même niveau que l'élément chauffant. Les sections restantes de la paroi du cylindre sont isolées au mieux. Cela dit, il n'existe pratiquement aucun flux thermique considérable traversant ces parties restantes du système physique. Les deux actionneurs introduits sont contrôlés au moyen

d'un algorithme commun. Par conséquent, un seul actionneur est activé à la fois selon la valeur calculée de la commande.

5.1.2. Optimisation des paramètres de contrôle

La conception du contrôle pour la réalisation de la boucle de régulation présentée à la figure 5.1 consiste à définir les paramètres caractéristiques de l'algorithme de contrôle. Le contrôleur PID choisi pour la stratégie de contrôle impose la configuration de trois paramètres K_p , K_d , K_i . La configuration finale attribuée au contrôleur doit permettre d'obtenir un système dont le comportement est conforme aux critères d'évaluation suivants :

- un dépassement inférieur à 1.5 °C ;
- un temps de réponse inférieur à 2 min pour une température initiale pièce (25 °C) ;
- une erreur en régime permanent inférieure à 0.2 °C.

La méthode *Ziegler-Nichol* [17] permet d'estimer les gains optimaux pour obtenir un résultat de contrôle acceptable et conforme aux critères d'évaluation cités antérieurement. Cette méthode fixe les gains du contrôleur PID en fonction de deux autres paramètres K_c et P_c . Le premier paramètre correspond au gain proportionnel qui induit le système dans un état oscillatoire continu et non amorti. Le deuxième paramètre se rapporte à la période de ces oscillations. Leur estimation s'effectue en appliquant un contrôle proportionnel uniquement. La figure 5.3 illustre le cas de figure qui correspond au comportement critique du système. La figure 5.4 constitue un agrandissement d'une sous section de la figure 5.3. Elle illustre la mesure du paramètre critique P_c .

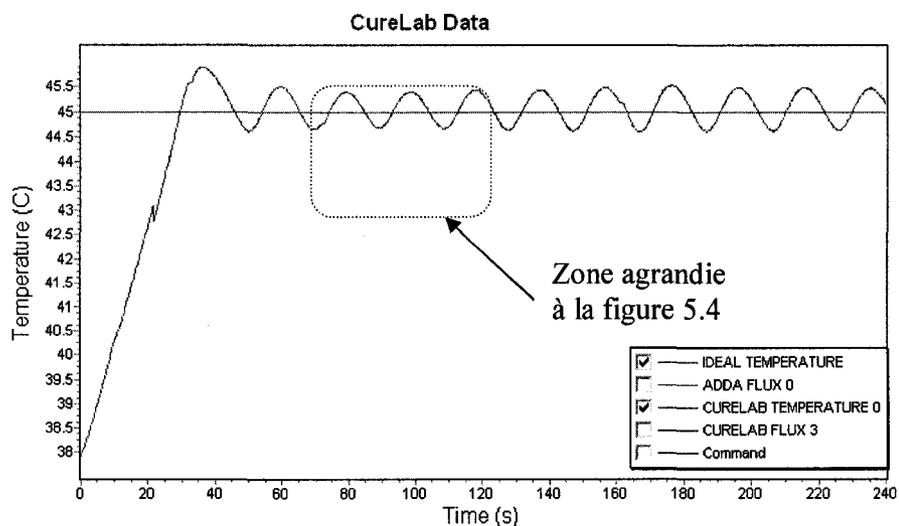


Figure 5.3 Évolution de la température acquise par le thermocouple du système cylindrique métallique contrôlée avec un contrôleur proportionnel avec gain critique

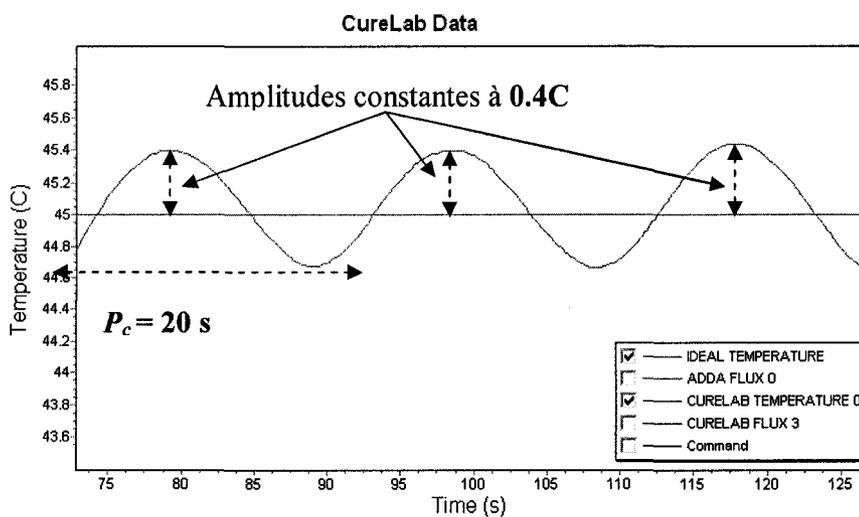


Figure 5.4 Agrandissement d'une zone de la figure 5.3 pour l'estimation de la période critique P_c

Les paramètres critiques ont été estimés à $K_c = 2.5$ et $P_c = 20$ s. Rappelons que le gain K_c est obtenu par itération d'essais jusqu'à l'observation de l'état oscillatoire illustré à la figure 5.3. La méthode de *Ziegler-Nichol* définit les gains du contrôleur PID [17] comme suit :

$$K_p = 0.6 \cdot K_c \quad (5.2)$$

$$K_i = \frac{2 \cdot K_p}{P_c} \quad (5.3)$$

$$K_d = \frac{K_p \cdot P_c}{8} \quad (5.4)$$

La figure 5.5 illustre les résultats d'un essai de régulation thermique sous contrôle par PID pour les gains estimés de la méthode de *Ziegler-Nichol*, à savoir $K_p = 1.5$, $K_i = 0.15$ et $K_d = 3.75$.

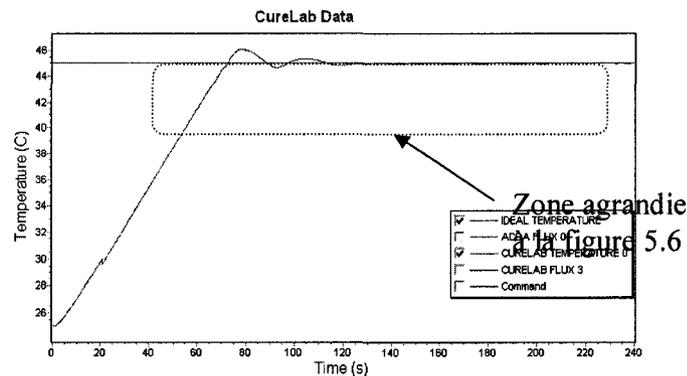


Figure 5.5 Évolution de la température acquise par le thermocouple du système cylindrique métallique contrôlée avec un contrôleur PID configuré selon la méthode de *Ziegler-Nichol*

La figure 5.6 représente un agrandissement d'une zone de la figure 5.5. Elle permet de quantifier les grandeurs d'évaluation de la qualité du contrôle.

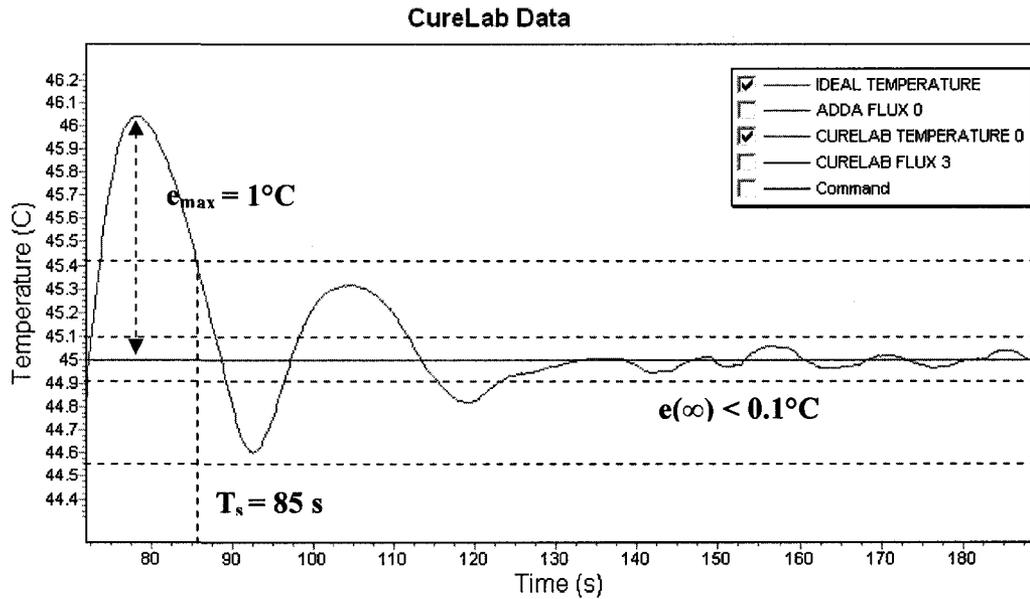


Figure 5.6 Agrandissement d'une sous-zone de la figure 5.5 pour l'estimation des critères d'évaluation du contrôle.

Notons que les résultats obtenus au niveau de cette configuration convergent vers les performances recherchées, soit celles qui présentent une conformité aux critères d'évaluation imposés. Les gains obtenus seront considérés au niveau des essais expérimentaux exposés dans les sections à venir.

5.2. Application du système DAC pour la caractérisation industrielle d'une résine thermodurcissable

5.2.1. Configuration de l'application

Le système DAC réalisé dans le cadre de ce mémoire est appliqué à la caractérisation thermique industrielle de résines thermodurcissables. Ces résines possèdent la propriété de polymériser sous l'effet de la température. Le phénomène de polymérisation produit de l'énergie thermique. La quantification de cette énergie et de son évolution temporelle présentent un intérêt considérable dans les applications industrielles dans la mesure où elle s'avère être porteuse d'informations pertinentes permettant de caractériser ledit matériau.

Le système illustré au niveau de la figure 5.1 représente un moule métallique destiné à recevoir la résine à caractériser. La figure 5.7 montre la configuration du système étudié qui contient la résine thermodurcissable à analyser.

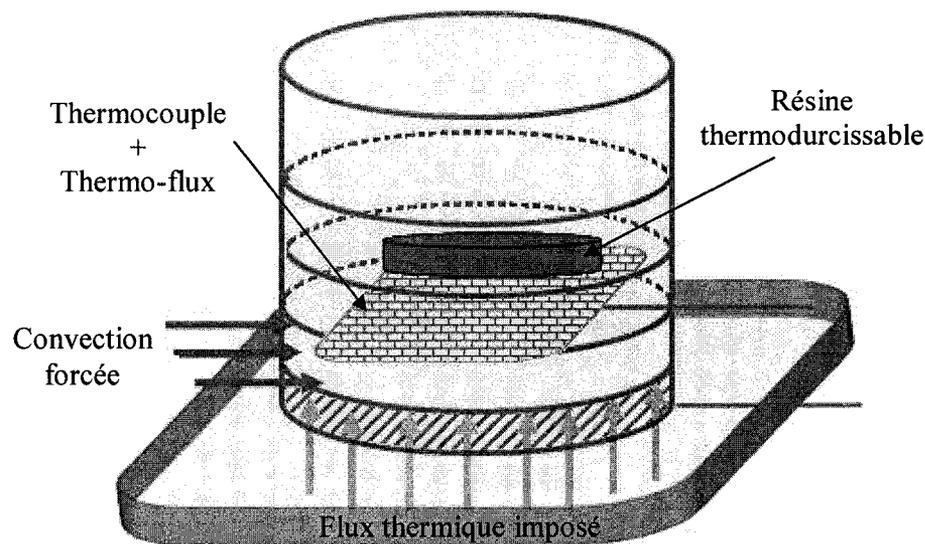


Figure 5.7 Configuration du moule étudié avec contenant la résine thermodurcissable

La caractérisation de la résine requiert des conditions adiabatiques au moment du déclenchement de la polymérisation de la résine. Cela est nécessaire pour permettre une détection complète de l'enthalpie de réaction générée au cours de la polymérisation. Pour cela, le cylindre est isolé thermiquement de sorte à éliminer tout échange thermique avec l'environnement extérieur. Pratiquement, il reste difficile d'obtenir de telles conditions. Toutefois, il demeure possible d'imposer des pertes thermiques constantes en fonction des caractéristiques du moule étudié. Ces pertes peuvent être quantifiées et seront cumulées aux valeurs d'enthalpie mesurées lors du processus de polymérisation de la résine dans le but d'approximer l'enthalpie de polymérisation.

5.2.2. Résultats

Les essais réalisés ont porté sur la formulation d'une résine à base de vinylester. Pour une certaine quantité massique de résine liquide, la formulation comprend les proportions massiques suivantes : l'initiateur CHP-50 (1.5 phr), l'accélérateur (DMA) (0.04 phr) et le promoteur CoNap6 (0.1 phr). L'essai de caractérisation a été effectué dans des conditions isothermes fixées à une température de 30 °C. La figure 5.8 montre les résultats obtenus. Remarquons la stabilité de la température mesurée tout au long de l'essai de polymérisation. Ce n'est pas le cas pour le flux thermique, qui illustre bien la nature exothermique de la réaction de polymérisation. La stabilisation du flux à une valeur non-nulle à la fin de l'expérience indique l'existence de pertes thermiques constantes dans le moule étudié.

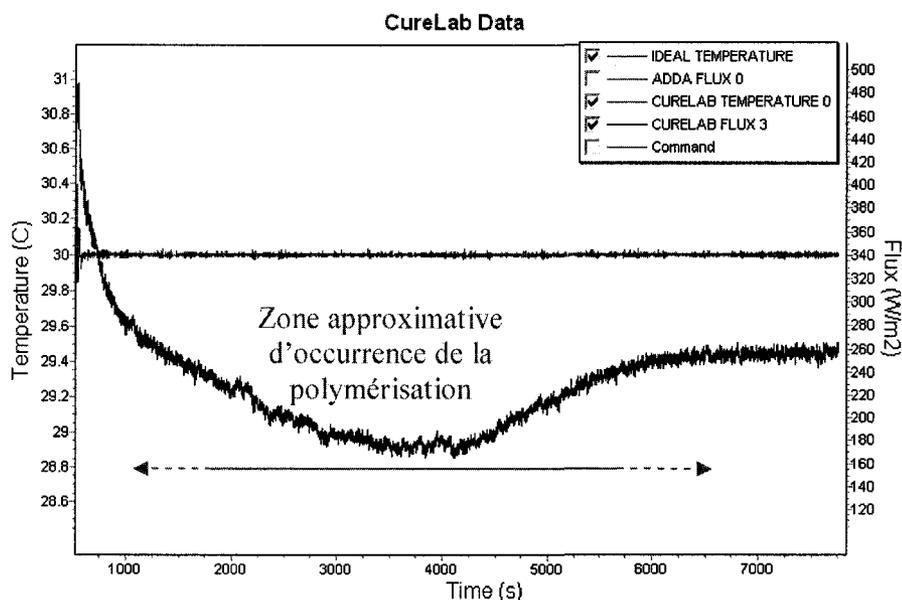


Figure 5.8 Évolution de la température et du flux thermique en fonction du temps pour un essai de polymérisation d'une résine vinylester

La figure 5.9 permet de comparer les résultats obtenus au cours de l'essai de polymérisation précédent avec ceux obtenus dans le cas du chauffage d'une résine préalablement polymérisée.

Remarquons l'absence de chute temporaire du flux thermique au cours de l'essai illustré à la figure 5.9. Un tel résultat était prévisible puisqu'aucun phénomène générateur d'énergie thermique ne se produit. La variation initiale du flux est due à la période de stabilisation pendant laquelle la température du moule devient homogène. Cette remarque s'applique également à l'essai de polymérisation de la figure 5.8. Ceci complique l'identification de l'instant du déclenchement de la polymérisation dans le but de pouvoir quantifier l'enthalpie de réaction.

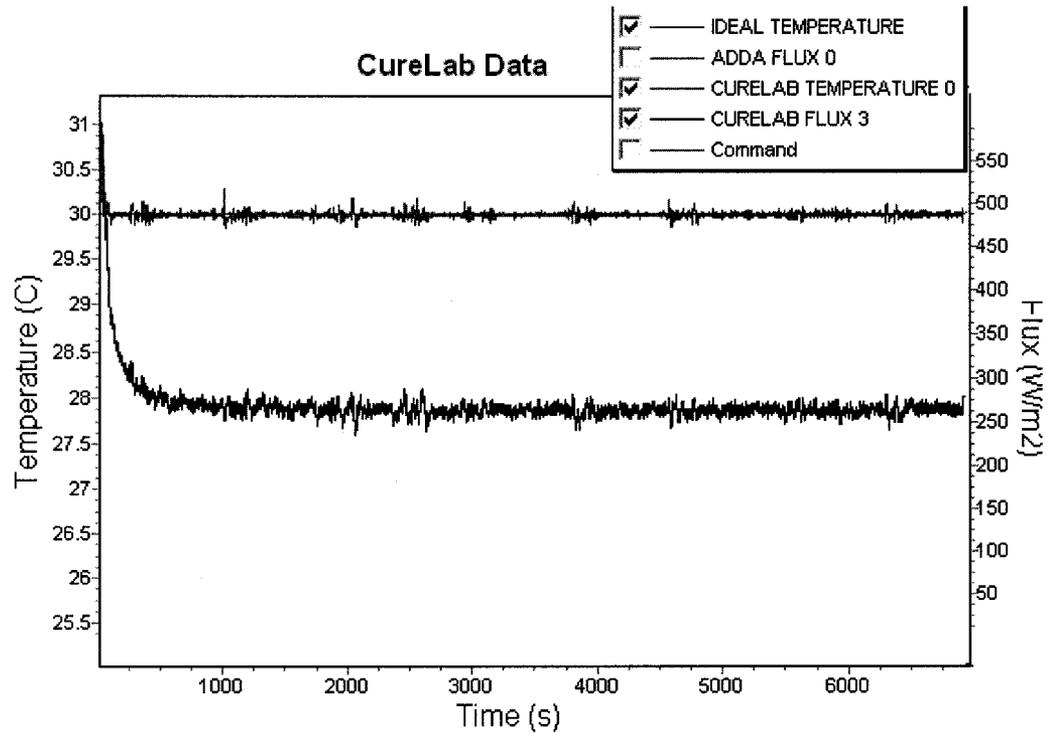


Figure 5.9 Évolution de la température et du flux thermique en fonction du temps pour le chauffage d'une résine vinylester polymérisée.

Il reste possible d'approximer le flux thermique associé spécifiquement à l'enthalpie de polymérisation en calculant des flux thermiques détectés pendant les essais de polymérisation et après polymérisation des figure 5.8 et 5.9 respectivement. Cela donne par soustraction le flux illustré à la figure 5.10. Notons que le flux de polymérisation illustré a été filtré numériquement selon la méthode des moindres carrés. Cela permet d'éliminer les irrégularités du signal dues au bruit d'acquisition.

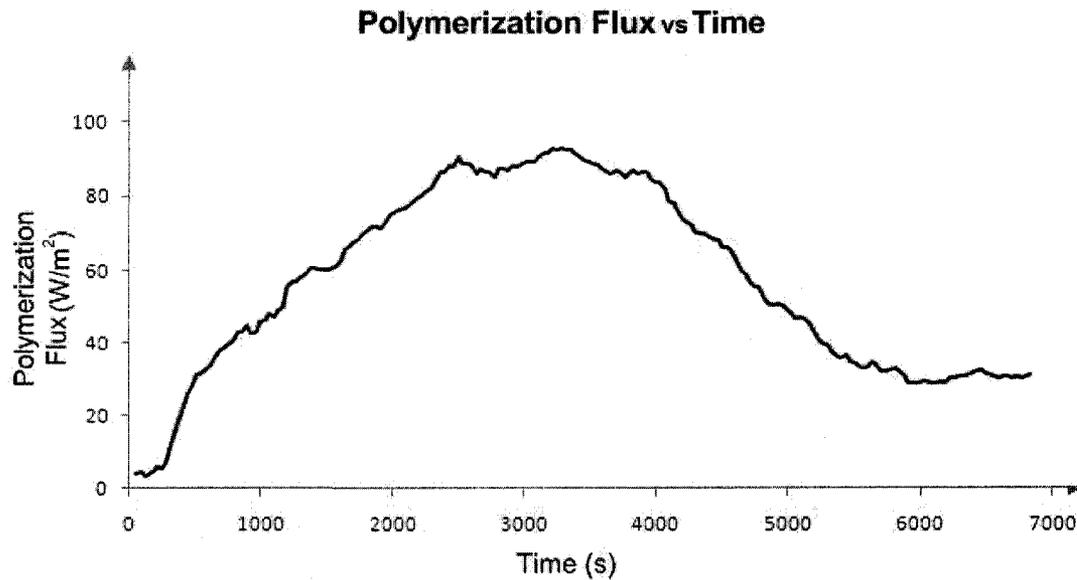


Figure 5.10 Évolution du flux thermique de polymérisation après filtrage en fonction du temps pour une résine vinylester

Notons que les conditions de températures initiales sont différentes dans les deux essais illustrés au niveau des figures 5.8 et 5.9 respectivement. Cela explique la valeur non nulle de la différence du flux thermique à la phase finale de l'essai tel qu'illustré à la figure 5.10.

La détermination de l'évolution du degré de cuisson durant la polymérisation s'effectue à partir du flux de polymérisation de la figure 5.10. En effet, l'équation qui permet de calculer le degré de cuisson $\alpha(t)$ est la suivante :

$$\alpha(t) = \frac{\int_0^t F(t) dt}{\int_0^{\infty} F(t) dt} \quad (5.5)$$

où $F(t)$ représente le flux thermique de la polymérisation. Concrètement, puisque la période d'échantillonnage T du flux thermique est constante, le degré de cuisson peut être approximé par la relation suivante :

$$\alpha(kT) = \frac{\sum_{i=0}^k F(iT)}{\sum_{i=0}^m F(iT)} \quad (5.6)$$

où m est le nombre total des échantillons de flux acquis et $k < m$. De ce fait, l'évolution du degré de cuisson de la résine utilisée est mesurée dans le temps durant un essai de polymérisation à la figure 5.11.

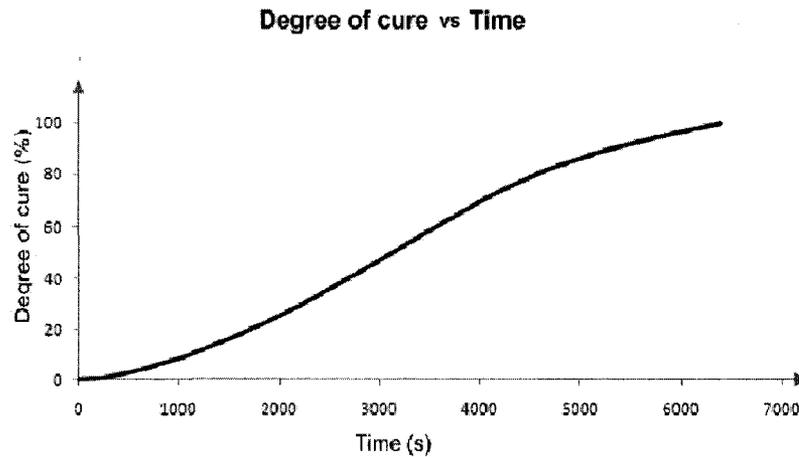


Figure 5.11 Évolution du degré de polymérisation dans le temps pour une résine vinylester

Plusieurs analyses pertinentes peuvent être effectuées sur l'évolution du degré de polymérisation. Cela permet de déterminer plusieurs grandeurs caractéristiques de la résine étudiée.

5.2.3. Conclusion

La caractérisation thermique d'une résine thermodurcissable dans un environnement industriel est une application du système d'acquisition et de contrôle développé dans le cadre de cette étude. La capacité du système DAC à détecter les variations de flux thermique caractéristiques du phénomène de polymérisation d'une résine thermodurcissable a été établie. Cela reste aussi révélateur de la qualité du contrôle thermique qu'offre le système DAC dans le sens où la température du moule reste maintenue à une valeur fixe pendant toute la manipulation.

CONCLUSIONS ET PERSPECTIVES

Un nouveau système d'acquisition et de contrôle pour la caractérisation thermique des résines polymères, a été présenté dans le présent travail. Ce système a été particulièrement appliqué à l'acquisition et au contrôle thermique dans le but de produire un outil de caractérisation industrielle de résines thermodurcissables utilisées dans la fabrication des composites à haute performance. Ce nouveau système offre l'opportunité d'une collaboration virtuelle basée sur l'échange d'informations et de données à travers les réseaux informatiques. Aussi, la possibilité d'un contrôle à distance des procédés industriels est rendue possible par le biais des modèles et applications logicielles conçues et implémentées.

L'aspect réparti du système va dans la direction des tendances technologiques actuelles qui visent à exploiter l'omniprésence des infrastructures des réseaux informatiques amplement répandues et considérablement populaires. Par ailleurs, le caractère multidisciplinaire de l'industrie des composites crée le besoin de recourir à des systèmes similaires dans le but d'améliorer aussi bien la collaboration des acteurs intervenant dans le secteur, que les processus de supervision et du contrôle des procédés qui se rattachent à ladite industrie.

La conception modulaire de l'architecture logicielle du système DAC permet une réutilisation des ressources et offre une capacité d'évolution de la solution proposée. Le développement de cette solution repose sur les principes de la conception et de l'implémentation orientées-objets d'un côté, et l'imprégnation des patrons de conception logicielle les plus en vigueur d'un autre côté. Également, l'emphase a été mise sur la généralité des concepts créés dans le but de permettre une extension facile des fonctionnalités d'acquisition et de contrôle du logiciel à des procédés de natures variées.

Le support réseau a été élaboré en s'appuyant sur les principes de l'architecture orientée-services. Effectivement, l'ensemble des fonctionnalités relatives à l'acquisition des données et au contrôle, ainsi que les activités qui s'y rattachent, est implémenté sous la forme d'entités logicielles invocables à distance. L'utilisation d'une telle approche a permis d'en évaluer l'efficacité au niveau de la réalisation des systèmes logiciels répartis œuvrant dans les domaines du contrôle industriel. Il faut rappeler que les développements actuels reposent principalement sur les méthodes classiques de répartition des systèmes, fondées sur les mécanismes de communication bas-niveau, ou encore sur les technologies web dans le cas de la supervision des procédés. L'adoption des services localisés à distance comme choix technologique a donné lieu à des performances satisfaisantes et a permis de renforcer la vision modulaire globale de l'architecture du logiciel conçu.

La conception de la partie matérielle du système DAC a pris en compte les différentes contraintes qualitatives qui se rapportent au domaine d'application visé. Cela touche à la précision de la conversion des signaux extraits du procédé et à la précision de l'application des commandes de contrôle. Le support de la communication avec le logiciel développé prend en considération les situations de désynchronisation et des erreurs de transmission afin d'obtenir un niveau de robustesse satisfaisant. L'application du système à la caractérisation thermique des résines thermodurcissables a permis de valider le fonctionnement global de l'application. Les résultats obtenus concordent avec les objectifs énoncés au départ, et s'accordent avec les hypothèses préalablement posées.

Les essais réalisés ont permis aussi de déceler des points d'amélioration à considérer dans le cadre de travaux futurs. En effet, il serait possible de découpler le contrôle des actionneurs afin de donner la possibilité d'en combiner les effets simultanément. Par ailleurs, il serait intéressant d'implémenter un protocole plus élaboré pour la communication entre les parties logicielle et matérielle. Il s'agirait, entre autres, de découpler les requêtes d'acquisition de données et les commandes de contrôle. Sur un

autre plan, le conditionnement analogique des signaux acquis pourrait contribuer à réduire les effets du bruit. Enfin, il serait possible d'accroître la précision des résultats en améliorant les propriétés d'isolation thermique du moule de caractérisation.

RÉFÉRENCES

- [1] G. Coulouris. "Distributed Systems, Concepts and Design " International Journal of Production Research, , 0-321-26354-5, 2005.
- [2] Puder, Arno, Kay Rmer, and Frank Pilhofer. "Chapter 2 - Basic Concepts". Distributed Systems Architecture: A Middleware Approach. Morgan Kaufmann Publishers. 2006.
- [3] Kircher, Michael, and Uwe Zdun. "Chapter 1 - Introduction To Distributed Systems". Remoting Patterns: Foundations of Enterprise, Internet and Realtime Distributed Object Middleware. John Wiley & Sons. 2005.
- [4] A. Chaffee and B. Martin. "Module 3 (B)- Architecture des systèmes répartis". *Notes de cours*, École Polytechnique de Montréal, 2007.
- [5] Alejandro Quintero. "Introduction to CORBA". *JGuru Training by the Magelang Institute* , 1999.
- [6] W. D. Li and Z. M. Qiu, "State-of-the-art technologies and methodologies for collaborative product development systems" International Journal of Production Research, 44:13, 2525 2559 2006.
- [7] S. Krakowiak. "Middleware Architecture with Patterns and Frameworks" Creative Commons Licence, 2008.

- [8] Asbury, Stephen, and Scott R. Weiner. "Chapter 13 - Introduction to Java RMI". *Developing Java Enterprise Applications*. John Wiley & Sons. © 1999.
- [9] Rammer, Ingo, and Mario Szpuszta. "Chapter 2 -.NET Remoting Basics". *Advanced .NET Remoting, Second Edition*. Apress. © 2005
- [10] C. D. Schmidt, "C++ network programming. Mastering complexity with ACE and patterns", *Composite Structures*, vol. 1, Editions Addison-Wesley, Toronto, 2002.
- [11] J. Kurumbalapitiya and H. Hoole , "Data Acquisition" *The Electrical Engineering Handbook*, Ed. Richard C. Dorf, 2000.
- [12] R. Oppliger "Chapter 1 - Introduction". *Security Technologies for the World Wide Web, Second Edition*. Artech House. © 2003.
- [13] J. Curtis, " Mechatronics: an Introduction" *Taylor & Francis Group, LLC*. 2006.
- [14] C. Anderson, "Mechatronics System Control, Logic, and Data Acquisition" *Taylor & Francis Group, LLC*. 2008.
- [15] J. Kurumbalapitiya, S. Ratnajeevan, H. Hoole "Computers, Software Engineering, and Digital Devices" *CRC Press LLC*, 2000.
- [16] R. Natarajan, "Computer-Aided Power System Analysis" *by Marcel Dekker*, 2002.

- [17] J. O'Shea, "Introduction à la commande par ordinateur," *Notes de cours*, Presses internationales Polytechnique 1987.

- [18] R. Gourdeau et J. Thistle, "Introduction à la commande des systèmes linéaires" *Notes de cours*, Chapitre 4, 2008.

- [19] L. Wang and S. Liao, "Distributed Supervisory Control and Data Acquisition (SCADA) software: Issues and state of the art ", 0-7803-8449-0, 2004.

- [20] R. Iv. Arnaudov and I. N. Dochev, "Functional Generator and Data Acquisition System Controlled by Internet", 0-7803-9446-1, 2005.

- [21] G. Zhu, "A Software Component for Network Based Data Acquisition And Control Applications", 1-4244-0686-2, 2006.

- [22] Analog Devices, "MicroConverter® Multichannel 24-/16-Bit ADCs with Embedded 62 kB Flash and Single-Cycle MCU", *DataSheet*.

- [23] J. D. Gibson, "The communications handbook ", CRC Press LLC, 2002.

- [24] Omega "HFS-1, HFS-2, HFS-3, HFS-4 Thin Film Flux Sensors", User guide.

ANNEXES

ANNEXE A : Diagramme de cas d'utilisation du système logiciel d'acquisition et de contrôle

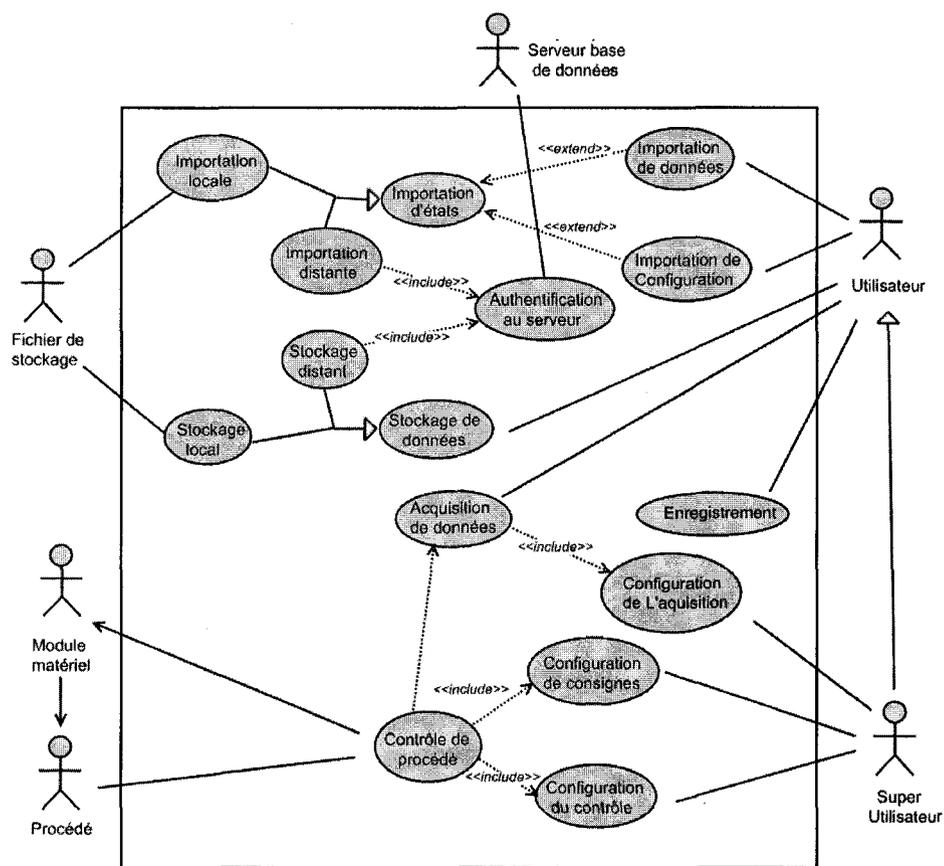


Figure A.1 : Diagramme de cas d'utilisation du système d'acquisition et de contrôle développé

ANNEXE B : Spécifications et schéma du microcontrôleur ADuC845 [22]

Features

High resolution Σ - Δ ADCs
 2 independent 24-bit ADCs on the ADuC845
 Single 24-bit ADC on the ADuC847 and single 16-bit ADC on the ADuC848
 Up to 10 ADC input channels on all parts
 24-bit no missing codes
 22-bit rms (19.5 bit p-p) effective resolution
 Offset drift 10 nV/°C, gain drift 0.5 ppm/°C chop enabled

Memory

62-kbyte on-chip Flash/EE program memory
 4-kbyte on-chip Flash/EE data memory
 Flash/EE, 100-year retention, 100 kcycle endurance
 3 levels of Flash/EE program memory security
 In-circuit serial download (no external hardware)
 High speed user download (5 sec)
 2304 bytes on-chip data RAM

8051-based core

8051-compatible instruction set
 High performance single-cycle core
 32 kHz external crystal
 On-chip programmable PLL (12.58 MHz max)
 3 \times 16-bit timer/counter
 24 programmable I/O lines, plus 8 analog or digital input lines
 11 interrupt sources, two priority levels
 Dual data pointer, extended 11-bit stack pointer

On-chip peripherals

Internal power-on reset circuit
 12-bit voltage output DAC
 Dual 16-bit Σ - Δ DACs
 On-chip temperature sensor (ADuC845 only)
 Dual excitation current sources (200 μ A)
 Time interval counter (wake-up/RTC timer)
 UART, SPI®, and I2C® serial I/O
 High speed dedicated baud rate generator (incl. 115,200)
 Watchdog timer (WDT)
 Power supply monitor (PSM)
 Power
 Normal: 4.8 mA max @ 3.6 V (core CLK = 1.57 MHz)
 Power-down: 20 μ A max with wake-up timer running
 Specified for 3 V and 5 V operation
 Package and temperature range:
 52-lead MQFP (14 mm \times 14 mm), -40°C to +125°C
 56-lead LFCSP (8 mm \times 8 mm), -40°C to +85°C

Applications

Multichannel sensor monitoring
 Industrial/environmental instrumentation
 Weigh scales, pressure sensors, temperature monitoring
 Portable instrumentation, battery-powered systems
 Data logging, precision system monitoring

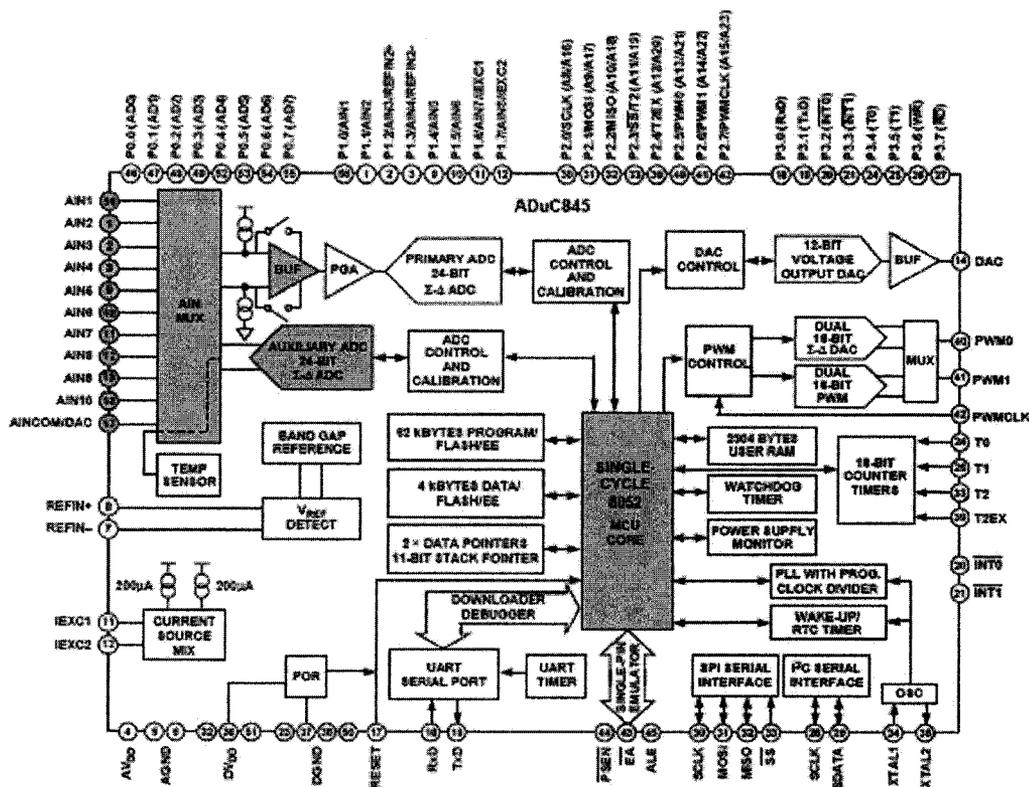


Figure B.1 : Schéma bloc détaillé du microcontrôleur ADuC845 [22]