Méthodes statistiques pour la prédiction de température dans les composants hyperfréquences

Grégory Mallet

Institut National des Sciences Appliquées

Thèse de doctorat

par Grégory MALLET

pour obtenir le grade de

Docteur de l'Institut National
des Sciences Appliquées de Rouen

Discipline : Informatique

Méthodes statistiques pour la prédiction de température dans les composants hyperfréquences

Statistical methods for temperature prediction in hyperfrequency components

Jury :

Gérard Bloch (Rapporteur)	Professeur à l'ESSTIN (Nancy)
Stéphane Canu (Directeur)	Professeur à l'INSA de Rouen
Emmanuel Duflos (Rapporteur)	Professeur à l'Ecole Centrale de Lille
Gilles Gasso (Encadrant)	Maître de conférence à l'INSA de Rouen
Philippe Leray	Professeur à Polytech'Nantes
Alain Rakotomamonjy	Professeur à l'Université de Rouen
Francois Requillard (Invité)	Ingénieur Thales Air System

Laboratoire d'Informatique, de Traitement de l'Information et des Systèmes (LITIS) - INSA de Rouen
Avenue de l'Université, 76801 Saint Etienne du Rouvray Cedex

25 Octobre 2010

Remerciements

Je tiens tout d'abord à remercier les rapporteurs de cette thèse, Gérard Bloch et Emmanuel Duflos, pour avoir eu la patience de relire ce manuscrit et pour leurs remarques et conseils qui m'ont permis de l'améliorer, même s'il faut me résoudre à ce qu'il ne soit jamais parfait.

Je tiens également à remercier Francois Requillard pour l'intérêt qu'il a porté à mon travail. J'espère qu'ainsi ce manuscrit pourra être diffuser dans les différents sites de Thales qu'il pourrait intéresser.

Je remercie aussi Alain Rakotomamonjy pour avoir accepté d'être le président de mon jury et avoir donc dû lire ce manuscrit.

Je remercie chaleureusement Stéphane Canu, pour avoir accepté d'être mon directeur de thèse et pour toujours avoir su voir ce qu'il y avait de positif dans mon travail, même lorsque moi-même je n'y croyais plus.

Il me faut évidemment remercier aussi mes deux encadrants. Philippe Leray, qui m'a encadré durant les deux premières années de cette thèse, que je remercie infiniment pour toutes les discutions intéressantes et fructueuses que nous avons eues et pour son écoute face à mes doutes et mes questions. Je ne remercierai également jamais assez Gilles Gasso, qui a pris le relai et m'a encadré ensuite jusqu'à la fin, pour avoir su m'orienter vers de nouvelles solutions et bien sûr pour son indulgence et son aide durant la très longue phase de rédaction.

Je remercie aussi Hubert Polaert pour avoir été mon lien avec le site de Thales Ymare et pour son ouverture d'esprit vis-à-vis des solutions qu'un mathématicien peut apporter aux problèmes rencontrés en thermique. Je n'oublie pas Clément Tolant et Philippe Eudeline qui m'ont accueilli avec sympathie et m'ont permis de m'intégrer au sein de cette entreprise. A travers eux, je veux également remercier toutes les personnes de Thales Air Systems (Ingénieurs, techniciens ou stagiaires) que j'ai pu cotoyer. Les réussites de cette thèse tiennent aussi à leur aide et à ma bonne intégration dans leur entreprise.

Je ne peux pas non plus oublier tous ceux dont j'ai partagé le bureau au laboratoire LITIS et même au laboratoire PSI au gré de mes déménagements. Olivier, Sam et Stijn, les bayésiens du bout du monde, Firas que j'ai rencontré aussi là-bas. Puis Frédéric et Karina qui m'ont permis de me rapprocher un peu plus de la vie du laboratoire. Je tiens particulièrement à remercier Karina pour sa bonne hummeur qui m'a beaucoup aidée. Bonne humeur qu'elle partage avec Elsa dont le rire résonne encore à mes oreilles. Enfin, ce fut le temps du Déménagement (le vrai, celui de l'INSA), et d'un bureau plus grand mais avec plus de gens (Aurélie, Carlo, Julien, Paul et Guillaume si vous lisez ce texte, c'est que vous ne travaillez pas vraiment).

Bien sûr, je salue également tous les autres thésards du laboratoire dont j'ai pu croiser la route, Gaëlle, Vincent, Benjamin, Rémi (avec un i), et Xilan. La vie d'un thésard serait évidemment bien triste si d'autres ne souffraient pas en même temps. Je salue aussi tout personnel du LITIS pour m'avoir accueilli au sein de ce laboratoire et plus particulèrement Brigitte, Sandra et Florence, les secrétaires, sans qui départements et laboratoires s'écrouleraient je pense. Je salue aussi Sébastien sa sympathie et pour avoir toujours su résoudre les problèmes que j'ai pu rencontrer.

Enfin, je remercie tous ceux extérieurs au monde de la recherche, amis et famille, qui ont eu à me cotoyer, surtout pendant la rédaction. Même s'ils ne pourront jamais tout à fait comprendre ce que représente la rédaction d'une thèse de doctorat, ils m'ont toujours supporté (dans les deux sens du terme) tout au long de cette périlleuse aventure. A ce titre, mes parents auront sans doute eu le plus à subir, donc je les remercie une nouvelle fois.

Table des matières

Notations

Thermique Notations des variables et des constantes thermiques.

a ..... Effusivité
c ..... Célérité
c_m ..... Chaleur thermique massique
c_v ..... Chaleur thermique volumique
$cste$ ..... Constante quelconque
E ..... Énergie
$\overrightarrow{grad}$ ..... Gradient
g ..... Gravité
h_e ..... Coefficient d'échange convectif
h ..... Constante de Planck
k ..... Conductivité
k_B ..... Constante de Boltzmann
P ..... Puissance thermique
p ..... Pression
$\overrightarrow{q}$ ..... Flux de chaleur
r ..... Réflectance
S ..... Surface
T ..... Température
t ..... Temps
U ..... Énergie interne
V ..... Volume
α ..... Absorptance
λ ..... Longueur d'onde d'un rayonnement
μ ..... Viscosité
ρ ..... Masse volumique
ϕ ..... Fonction de base dans une méthode à éléments finis
σ ..... Constante de Stefan-Boltzmann
τ ..... Transmittance

Statistiques Notations utilisées pour la définition des espaces, des données et des fonctions.

$\mathscr{H}$ ..... Ensemble d'hypothèses, la plupart du temps, il s'agira d'un espace de Hilbert.
$\mathscr{F}$ ..... Famille de fonctions
$\mathbb{R}$ ..... Espace des réels
$\mathbb{N}$ ..... Espace des entiers naturels

$\mathscr{X}$ ..... Espace d'entrée pour les données exemples
$\mathscr{Y}$ ..... Espace de sortie pour les données exemples
$\mathscr{P}$ ..... Famille de lois de probabilité
$\mathbb{P}$ ..... Loi de probabilités
$\mathbb{E}$ ..... Espérance
$\mathbb{V}$ ..... Variance
L ..... Rapport de vraisemblance
$\mathbf{X} \in \mathbb{R}^{n \times d}$ ..... Ensemble de vecteurs de données
$\mathbf{x} \in \mathbb{R}^d$ ..... Vecteur de données
$x_i \in \mathbb{R}$ ..... $i_{\text{ème}}$ valeur d'un vecteur de donnée
$y \in \mathbb{R}$ ..... Vecteur de sortie
y_i ..... $i_{\text{ème}}$ sortie
$\mathbf{w}$ ..... Vecteur de paramètres d'un modèle
w ..... Paramètre d'un modèle
$\mathbf{1}^d$ ..... Vecteur de taille d contenant des 1
α, β ..... Multiplicateurs de Lagrange
$\mathscr{L}$ ..... Lagrangien
ℓ ..... Fonction coût
$R(.)$ ..... Risque
$R_e(.)$ ..... Risque empirique
$K(.,.)$ ..... Fonction noyau entre deux variables
$\text{sign}(.)$ ..... Fonction signe
$f(.)$ ..... Fonction de décision
$\underset{x \in \mathscr{X}}{\text{argmin}}\, T(.)$ ..... Valeur de x qui donne le minimum de $T(.)$
$\underset{x \in \mathscr{X}}{\text{argmax}}\, T(.)$ ..... Valeur de x qui donne le maximum de $T(.)$
$min(.,.)$ ..... Fonction retournant le minimum entre deux variables
$max(.,.)$ ..... Fonction retournant le maximum entre deux variables
$\langle .,. \rangle_{\mathscr{H}}$ ..... Produit scalaire dans l'espace $\mathscr{H}$
$V_\upsilon(\mathbf{x})$ ..... Voisinage au point $\mathbf{x}$ d'une taille υ
$d(.,.)$ ..... Fonction de distance
$\mathscr{O}(.)$ ..... Domination asymptotique, représente la complexité
σ ..... Largeur de bande
C ..... Pondération des données mal classées pour le classifieur SVM
w_0 ..... Ordonnée à l'origine d'un hyperplan
N ..... Taille de la base d'apprentissage
ε ..... Terme de régularisation
ξ ..... Variable de relâchement
$\mathbf{K}$ ..... Matrice de Gram d'un noyau

Acronymes Quelques acronymes utilisés au cours du manuscrit.

SVM ..... *Support Vector Machine* ou Séparateur à Vaste Marge
RKHS ..... *Reproducing Kernel Hilbert Space* ou Espace de Hilbert à Noyau Reproduisant
KKT ..... Karush-Kuhn-Tucker
RBF ..... *Radial Basis Function* ou Fonction à Base Radiale
FEM ..... *Finite Element Method* ou Méthode à Éléments finis
IR ..... Infra Rouge
SVD ..... *Singular Value Decomposition* ou Décomposition en valeurs singulières

Résumé

Les Radar - pour RAdio Detection And Ranging - sont des instruments permettant de mesurer la position et la vitesse d'objets (avions, bateaux, pluie, ...) à l'aide d'ondes électromagnétiques. Les ondes utilisées appartiennent à la gamme des ondes radio (fréquence inférieure à 3000 GHz), compte tenu de leurs bonnes propriétés de transmission au sein de l'atmosphère. Le principe de fonctionnement commence par l'émission d'une onde la plus puissante possible par l'émetteur du radar. Cette onde est ensuite diffusée par l'antenne dans l'atmosphère. Pour qu'il y ait détection, l'onde doit alors être réfléchie par une cible puis captée par une antenne (qui peut être différente de l'antenne d'émission) et amplifiée pour devenir exploitable. La position est alors obtenue par le temps de parcours de l'onde et la vitesse par le déphasage en fréquence lié à l'effet Doppler. Pour les radars monostatiques (où les antennes d'émission et de réception sont confondues), un mode de fonctionnement pulsé est nécessaire puisqu'une même antenne ne peut émettre et recevoir en même temps. Dans le scénario le plus simple, l'onde électromagnétique va donc être émise durant un temps fixé qui sera suivi d'un « silence » du radar correspondant à la phase d'écoute et dont la durée est donnée par la distance maximale de détection du radar. Ce type de radar peut sommairement se décomposer en trois parties distinctes : une antenne, une partie analogique chargée de la génération de l'onde et de son amplification et une partie numérique qui est responsable du choix des scénarios d'impulsions ainsi que du traitement des données.

Les problèmes thermiques se situent principalement au niveau de la partie analogique du radar. En effet, l'un des paramètres les plus importants pour les performances de détection est la puissance de l'onde que le radar est capable d'émettre. En plus d'un générateur radio (qui va produire l'onde), ce sous-système contient donc de nombreux étages d'amplification pour obtenir l'onde la plus puissante possible. Ces étages sont réalisés à partir d'une série de transistors de puissance croissante. Ce sont sur ces composants que se localisent les densités de puissance les plus élevées. Or, du fait de l'effet Joule, une forte densité de puissance implique une forte puissance dissipée sous forme d'agitation thermique et donc une température élevée. L'architecture du radar est donc conçue pour étaler puis évacuer cette chaleur à l'aide de systèmes de refroidissement. Cependant, la tendance actuelle est à une augmentation constante de la puissance des radar. De plus, les systèmes de refroidissement utilisent le plus souvent de l'air ce qui constitue la solution la plus pratique et la plus fiable mais également la moins bonne solution en terme de performance. Le management thermique est ainsi devenu le principal facteur limitant de la puissance des radar et un paramètre non négligeable dans leur coût de fabrication et de fonctionnement.

Les systèmes de refroidissement atteignent aujourd'hui leurs limites et il faut se tourner vers d'autres méthodes pour continuer d'accroitre les performances des radar. Une solution envisageable est de mieux utiliser les marges intégrées à chaque étape de la conception. En effet, les spécifications des différents éléments du système de refroidissement et des composants électroniques sont établies en prévision du pire cas possible. Cette méthode implique des contraintes très fortes sur l'intégralité du fonctionnement du radar alors qu'elles ne sont utiles que dans les cas extrêmes. Toutefois, pour assouplir ces contraintes, il faut être capable de prédire la température de jonction du transistor en temps-réel pour estimer la marge en température disponible. De nombreux logiciels existent pour réaliser des modèles thermiques précis mais le coût de ces prédictions en terme de puissance de calcul et de temps nécessaire les rend inapplicables dans ce contexte. Ces problèmes sont connus depuis longtemps et un grand nombre d'algorithmes ont été développés pour obtenir des modèles plus compacts. La prédiction reste cependant totalement dépendante de la connaissance théorique disponible sur l'architecture du système de refroidissement et du composant. Une autre approche consiste à appliquer les méthodes de mesures de température existantes afin

de générer des bases de données représentatives des variations de température au cours du fonctionnement réel du radar. Ces données peuvent alors être utilisées pour apprendre un modèle thermique.

C'est dans ce cadre que cette thèse va s'intéresser à un cas simplifié des systèmes réels présents dans les radars. Le système étudié se limite à un seul composant monté sur un système de refroidissement réduit. Cette étude prend place au sein d'un projet régional réunissant Thales Air Systems, le laboratoire CORIA et le laboratoire LITIS où chaque acteur apporte son expertise. La problématique est en effet issue des études de Thales Air Systems qui produit des radar civils et militaires. La société apporte également au projet son expérience en matière de modélisation thermique et ses moyens de mesures de température pour les composants électroniques. Le CORIA, laboratoire en aérothermochimie, possède une très forte expérience en expérimentation et métrologie. Il est chargé de la réalisation d'une maquette instrumentée pour permettre l'acquisition d'une partie des données nécessaires. Enfin, les modèles statistiques développés au LITIS seront utilisés pour la dernière étape du projet. L'organisation de ce manuscrit reprend cet enchaînement naturel et est présentée ci-dessous.

Chapitre I – Modélisation thermique

Le premier chapitre est consacré à la présentation succincte des phénomènes thermiques et leur modélisation. Les principaux modes de transfert de l'agitation thermique ainsi que les lois physiques gouvernant ces transferts sont abordés. Toutefois, cette approche purement analytique se révèle inapplicable lorsque l'architecture du système thermique devient complexe. Une alternative est représentée par les méthodes de résolutions numériques qui sont présentées dans le chapitre. Ces méthodes sont à la base des logiciels de simulation thermique utilisés habituellement pour la prédiction de température.

Notre principale contribution dans le chapitre I est l'application de ces approches numériques pour modéliser le comportement de composants radar simplifiés à différentes échelles afin d'appréhender finement les spécificités du problème. La première modélisation basée sur les méthodes à éléments finis suppose un composant électronique placé sur une plaque munie d'ailettes soumis à des impulsions de puissance et refroidi par un flux d'air. Elle révèle l'existence de deux dynamiques : une dynamique rapide liée à la réponse thermique du composante et une dynamique lente traduisant une dérive de la température au cours du temps. La deuxième modélisation focalise sur une étude plus détaillée de la dynamique rapide du composant électronique et a permis d'analyser la non-linéarité de la réponse du composant en fonction de la variation de sa conductivité thermique. Une conclusion de ces modélisations est que le système thermique en étude est un système dynamique dont l'entrée est représentée par les impulsions de puissance envoyées aux composants électroniques, la sortie est la température interne et les variables « pertubatrices » matérialisées par le système de refroidissement.

Chapitre II – Mesures de température

Les simulations du chapitre I ont permis de cerner les ordres de grandeur (dynamiques, valeur maximale de la température interne, ...) du phénomène thermique à étudier. Malheureusement les temps de calcul nécessaires pour simuler le comportement du composant électronique sur une brève durée sont prohibitivement élevés. Il importe alors de compléter notre connaissance par des mesures effectuées sur un système réel et pour cela choisir les méthodes de mesure de température adaptées au problème.

La première partie de ce chapitre décrit un état de l'art des méthodes de métrologie de température. Un comparatif synthétisant les forces et faiblesses de chaque méthode en fonction des spécifications définies grâce aux simulations est ensuite dressé. On en déduit alors les choix de dispositif de mesurage thermique retenus. Ainsi, nous avons contribué au développement de la maquette instrumentée par le laboratoire CORIA en proposant le choix de capteurs thermocouples qui sont suffisants pour mesurer la dynamique lente. Une description de cette maquette et des expérimentations réalisées est alors présentée. Pour accéder à la dynamique rapide, il a été nécessaire de faire appel à des mesures infrarouges. Le problème posé par cette technique est qu'elle nécessite la connaissance ou la détermination de l'émissivité des matériaux pour fournir des mesures précises. Pour automatiser le recalage

de l'émissivité, nous avons proposé une procédure efficace de traitement des images infrarouges (mesures) basée sur des méthodes de traitement d'images. Cette contribution a fait l'objet d'un dépôt de brevet [MP09].

Finalement à l'issue de ce chapitre, nous avons fixé deux bases de mesures devant servir à l'élaboration de modèles statistiques. La première base est formée par les résultats de simulation numérique (dynamique rapide) complétés par l'adjonction aux mesures de température d'un bruit reflétant la réalité des mesures effectuées via les méthodes infrarouges. La deuxième base de données est issue des mesures effectuées sur la maquette expérimentale en essayant d'utiliser des scénarios d'entrées les plus variés possibles compte tenu des contraintes sur l'utilisation de la maquette.

Chapitre III – Étude et développement de modèles statistiques d'un système thermique radar

Ce chapitre développe les approches de modélisation de type boîte noire que nous avons testées pour modéliser le comportement des systèmes thermiques à partir des bases de mesures élaborées précédemment. Après une présentation succincte du principe d'apprentissage statistique, nous avons focalisé sur son adaptation à l'identification de modèles dynamiques des composants. Quatre familles de modèles ont été explorées :

- les modèles linéaires : les modèles auto-régressifs et à erreur de sortie (Output Error, OE) sont considérés et leurs performances en généralisation servent de résultats de base,
- les réseaux de neurones récurrents construits à partir d'une structure dynamique de type OE,
- les machines à vaste marge (SVM),
- les modèles bayésiens dont l'intérêt est de permettre la prédiction de la température avec un intervalle de confiance.

Pour chaque famille, les fondations théoriques et les problématiques d'identification du modèle afférent sont présentées. Les performances en généralisation des différents modèles sont évaluées sur des données de test sous la forme de l'erreur quadratique entre les mesures réelles et les sorties simulées. En effet, le recours à une sortie simulée (donc dépendant des sorties précédentes du modèle et des entrées) est justifié par le fait qu'en phase d'exploitation, il ne sera pas possible d'accéder à la mesure de la température interne des composants électroniques.

Une analyse comparative des résultats obtenus en termes de capacité de généralisation, de facilité de mise en oeuvre est ensuite conduite. Les résultats obtenus [MLH+06], [MLP07] sont satisfaisants et montrent que les données à disposition traduisent essentiellement un comportement linéaire des systèmes thermiques étudiés. Les méthodes non-linéaires de modélisation comme les réseaux de neurones apportent une légère amélioration des performances des modèles linéaires. La quasi exhaustivité de l'étude comparative des approches de modélisation statistique aux données thermiques constitue notre principale contribution dans ce chapitre.

Chapitre IV – Identification de représentations d'état stables

Une méthode de modélisation thermique présentée dans le chapitre I est la représentation d'état. En effet sous des hypothèses simplificatrices, le modèle thermique peut être ramené à un modèle d'état de grande dimension. D'autre part, dans le chapitre III, les difficultés d'adaptation des méthodes à noyaux à la modélisation des systèmes dynamiques de type erreur de sortie nous ont conduit à explorer la piste de la représentation d'état. Les techniques d'identification de ces modèles font essentiellement appel à des méthodes d'algèbre linéaire qui ne tiennent pas compte de la nécessité que les paramètres identifiés doivent correspondre à un modèle stable. Ce problème est exacerbé si le système réel est à la limite de la stabilité ou de dimension élevée. Dans ce dernier chapitre, nous avons alors mené des travaux plus théoriques visant à identifier des modèles d'état stables en intégrant au problème la contrainte de stabilité (liée au rayon spectral de la matrice de transition d'état). Nous avons proposé un algorithme de résolution [MGC08] (dans le cas linéaire) basé une technique existante d'échantillonnage de gradient afin d'accéder au gradient du rayon spectral par rapport aux paramètres du modèle. Les pistes pour l'extension au cas non-linéaire seront évoquées.

La thèse se termine avec des conclusions sur l'étude menée et des perspectives de poursuite du projet. Ainsi à des fins d'amélioration des modèles obtenus, il serait intéressant de constituer une base de mesures infrarouges afin de tester nos approches statistiques sur des températures relevées à des échelles de temps de l'ordre des impulsions hyperfréquences. L'automatisation de ces acquisitions de mesure apporteraient une souplesse d'utilisation du système de mesure. De plus, cette automatisation permettrait d'avoir une cartographie 2D de l'évolution spatiale de la température des composants électroniques et d'étendre la modélisation statistique à la caractérisation du champ de température.

Valorisation des résultats de la thèse

[MP09] Mallet, G. and Polaert, H. (2010). Procédé de recalage automatique d'une image infrarouge Brevet, 22 pages.

[MLP07] Mallet, G., Leray, P. and Polaert H. (2007). Méthodes statistiques et modèles thermiques compacts. In Actes de la Conférence Extraction et Gestion de la Connaissance, pages 213-214.

[MLH+06] Mallet, G., Leray, P., Polaert, H., Tolant, C. and Eudeline P. (2006). Dynamic Compact Thermal Model with Neural Networks for Radar Applications. In Proceedings of 12th International Workshop on Thermal Investigations of ICs and Systems, pages 118-122.

[MGC08] Mallet, G., Gasso, G. and Canu, S. (2008). New methods for the identification of a stable subspace model for dynamical systems. In IEEE Workshop on Machine Learning for Signal Processing, pages 432-437.

1

Modélisation thermique

1.1 Introduction

La problématique abordée dans cette thèse est la modélisation des variations de la température interne des composants électroniques radar par des approches statistiques sur la base de la connaissance de mesures entrée-sortie recueillies expérimentalement. Étant donné que cette température dépend non seulement de l'architecture du composant mais aussi du système de refroidissement qui lui est associé, une prédiction fiable nécessite de prendre en compte l'intégralité du parcours de l'énergie thermique. Pour une bonne appréhension du problème, il convient tout d'abord d'analyser les méthodes de modélisation thermique existantes. Ces méthodes reposent sur une analyse théorique des phénomènes impliqués dans le transfert de chaleur. Après avoir décrit ces phénomènes et leur mise en forme mathématique, nous étudierons l'approche classique qui consiste à utiliser les informations disponibles sur le système (dimensions, structure, composition, ...) pour établir les paramètres du modèle thermique. Nous verrons notamment que les champs de température recherchés doivent être solutions d'équations aux dérivées partielles, qui peuvent s'avérer très complexes voire insolubles analytiquement. C'est pourquoi nous étudierons ensuite les méthodes de résolution numérique qui constituent une alternative intéressante à l'approche analytique. Les logiciels utilisés pour implémenter ces modèles numériques seront ensuite présentés et nous permettront d'établir les principales caractéristiques des modèles thermiques. Le chapitre se conclut avec l'application de ces approches aux cas des composants électroniques et des systèmes de refroidissement qui nous intéressent.

1.2 Le transfert thermique

Cette section ne constitue qu'une courte introduction aux phénomènes thermiques, une analyse plus poussée pourra notamment être trouvée dans [Che99].

L'énergie totale d'un système thermodynamique peut être décomposée sous 3 formes :
- l'énergie cinétique macroscopique,
- l'énergie potentielle liée aux forces extérieures,
- l'énergie interne (qui regroupe l'intégralité des énergies microscopiques).

L'énergie interne est à son tour décomposable en énergie cinétique microscopique et en énergie potentielle liée aux forces d'interaction entre les particules. Au niveau macroscopique, l'énergie cinétique moyenne des particules du système est représentée par une valeur appelée température. La chaleur quant à elle définit le transfert de cette énergie thermique entre deux systèmes. Les échanges d'énergie thermique s'effectuent suivant trois modes de transfert :

– la conduction,
– la convection,
– et le rayonnement.

Le transfert d'énergie par chaleur se réalise généralement par une combinaison de ces différents modes. Toutefois, dans certains cas, un unique mode peut être présent ou largement dominant vis-à-vis des autres modes, qui sont alors négligés.

1.2.1 Transfert de chaleur par conduction

L'énergie thermique est liée à l'agitation des particules du système. La conduction caractérise le transfert de proche en proche de cette agitation entre les particules. Au sein d'un gaz, toutes les molécules sont animées par un mouvement de translation et par un mouvement de vibration ou de rotation interne pour les molécules. Lors des collisions, les particules voisines interagissent entre elles et s'échangent de l'énergie. D'un point de vue macroscopique, il s'effectue alors un transfert de chaleur. Dans un liquide, le transfert collisionnel s'effectue toujours pour chaque particule mais uniquement autour de sa position d'équilibre. Dans un solide, le phénomène est plus complexe. Pour les métaux, le transfert de chaleur est assuré par l'intermédiaire des électrons. Dans le cas d'un solide cristallin non-métallique, la conduction est essentiellement liée aux vibrations (appelées « phonons ») du réseau d'atomes. Enfin, pour les solides amorphes non métalliques, le transfert reste associé aux collisions entre les atomes.

Observé au niveau macroscopique, le flux thermique $\overrightarrow{q}$ (mesuré en $W.m^{-2}$) traversant perpendiculairement une surface unitaire S est lié au gradient de température $\overrightarrow{gradT}$ par la loi de Fourier [Fou22] :

$$\overrightarrow{q} = -k\overrightarrow{gradT} \tag{1.1}$$

Le terme k représente alors la conductivité thermique du matériau en $W.K^{-1}.m^{-1}$.

Cette relation permet d'établir l'équation dite de la chaleur. En effet, le premier principe de la thermodynamique établit que pour un corps au repos et de volume constant, la variation d'énergie interne n'est liée qu'aux échanges de chaleur et aux éventuelles sources de chaleur. Or l'énergie interne $U(t)$ à l'instant t d'un système de volume V est liée à la température $T(t)$ au même instant par l'équation :

$$U(t) = \int_V \rho c_m T(t) dV + cste$$

où ρ représente la masse volumique , c_m la capacité thermique massique à volume constant du milieu et $cste$ représente une constante quelconque. Alors, si l'on considère que l'ensemble des échanges se font par conduction et que la puissance dissipée est déterminée par la fonction P, on peut écrire :

$$\Delta U = \int_V \rho c_m \frac{\partial T(t)}{\partial t} dV = \int_V P dV - \oint_S \overrightarrow{q} d\vec{S}$$

où $d\vec{S}$ est le vecteur normal à la surface de V, dirigé vers l'extérieur.

En utilisant le théorème d'Ostrogradki [Che99] et l'expression du flux de chaleur lié à la conduction, l'équation devient :

$$\int_V \rho c_m \frac{\partial T(t)}{\partial t} dV = \int_V P dV - \int_V div(-k\overrightarrow{gradT}) dV$$

Le volume V étant quelconque, en posant $c_v = \rho c_m$, on en déduit l'équation de la chaleur pour un milieu isotrope :

$$c_v \frac{\partial T(t)}{\partial t} = P + div(k\overrightarrow{gradT}) \tag{1.2}$$

qui caractérise le transfert de chaleur par conduction.

1.2.2 Transfert de chaleur par convection

Le transfert de chaleur par convection intervient à la séparation entre une paroi et un fluide en mouvement. La convection résulte de la combinaison du transfert de chaleur par conduction (diffusion) et du transport d'énergie par l'écoulement des particules du fluide (advection). C'est ce transfert d'énergie par déplacement qui la distingue clairement de la conduction. On distingue deux types de convection :

- la convection forcée, où le fluide est mis en mouvement par une action externe,
- et la convection naturelle, où le fluide est mis en mouvement uniquement par la différence de densité liée au gradient de température et par la poussée d'Archimède.

La convection est donc fortement liée à la mécanique des fluides. Les équations qui servent à décrire le mouvement des fluides, dites équations de Navier-Stokes [Nav22], sont relativement complexes et font intervenir l'équation de bilan de la masse, de la quantité de mouvement et de l'énergie au sein de chaque volume élémentaire de l'espace. Ces équations se simplifient si l'on suppose que le fluide est incompressible et newtonien (viscosité constante). On obtient alors pour chaque direction x,y et z de l'espace (la gravité est supposée colinéaire à l'axe z) :

$$
\begin{aligned}
-\frac{\partial p}{\partial x}+\mu\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}+\frac{\partial^2 u}{\partial z^2}\right) &= \rho\left(\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial x}+v\frac{\partial u}{\partial y}+w\frac{\partial u}{\partial z}\right)\\
-\frac{\partial p}{\partial y}+\mu\left(\frac{\partial^2 v}{\partial x^2}+\frac{\partial^2 v}{\partial y^2}+\frac{\partial^2 v}{\partial z^2}\right) &= \rho\left(\frac{\partial v}{\partial t}+u\frac{\partial v}{\partial x}+v\frac{\partial v}{\partial y}+w\frac{\partial v}{\partial z}\right)\\
-\rho g-\frac{\partial p}{\partial z}+\mu\left(\frac{\partial^2 w}{\partial x^2}+\frac{\partial^2 w}{\partial y^2}+\frac{\partial^2 w}{\partial z^2}\right) &= \rho\left(\frac{\partial w}{\partial t}+u\frac{\partial w}{\partial x}+v\frac{\partial w}{\partial y}+w\frac{\partial w}{\partial z}\right)
\end{aligned}
$$

où p est la pression au sein du fluide, ρ la densité du fluide, g la gravité, μ la viscosité du fluide et u,v et w les vitesses selon les 3 directions de l'espace. Il est clair que les équations fluidiques et thermiques au sein du fluide sont liées si l'on prend en compte la variation de la densité et de la viscosité du fluide en fonction de la température. Lorsque les conditions sont fixées (vitesse et propriétés du fluide, configurations géométriques, ...), il est possible de calculer un coefficient d'échange h_e qui relie la quantité de chaleur échangée à l'aire de la surface considérée S et à la différence de températures ΔT entre le fluide et la paroi. Dans la plupart des cas, il est donc avantageux de modéliser la convection par un simple coefficient d'échange aux limites du solide considéré. Il s'agit de la loi de Newton :

$$q = Sh_e\Delta T \tag{1.3}$$

En fonction de la vitesse et de la nature du fluide, des dimensions de l'écoulement et de la rugosité des parois, il existe deux régimes d'écoulement. À faible vitesse, l'écoulement est laminaire et il est possible de déterminer la vitesse et la température du fluide en tout point. Ce régime se caractérise par un coefficient d'échange qui augmente avec la vitesse du fluide. Lorsque la vitesse de l'écoulement dépasse une certaine valeur, l'écoulement adopte un régime dit turbulent et seules les valeurs moyennes de vitesse et de température peuvent être connues. Le coefficient d'échange devient alors très élevé. Toutefois, la pression nécessaire en aval de l'écoulement pour maintenir un débit suffisant augmente aussi et de manière plus importante que le coefficient d'échange. Ce type de régime n'est donc pas employé habituellement dans le cadre d'un système de refroidissement aéraulique.

1.2.3 Transfert de chaleur par rayonnement

Le transfert par rayonnement se traduit par un échange de chaleur entre deux corps sous la forme d'un rayonnement électromagnétique. Compte tenu de la dualité onde-corpuscule, le rayonnement peut être vu à la fois comme une onde électromagnétique et comme un ensemble de photons. Cette dualité implique une quantification de l'énergie radiative émise par une unité fondamentale E définie par :

$$E = \frac{hc}{\lambda}$$

où h représente la constante de Planck, c la célérité de la lumière et λ la longueur d'onde du rayonnement. Pour continuer l'analyse du transfert radiatif, il est nécessaire de faire intervenir un corps théorique idéal, appelé « corps noir », dont le spectre électromagnétique ne dépend que de sa température. La définition d'un corps noir est donnée par 3 propriétés essentielles de ce corps théorique :

- Le corps noir absorbe toutes les radiations incidentes, quelques soient la longueur d'onde et la direction.
- Pour une température et une longueur d'onde données, aucune surface ne peut émettre plus que le corps noir.
- Bien que la radiation émise par un corps noir soit fonction de la température et de la longueur d'onde, elle est indépendante de la direction. Le corps noir est un émetteur diffusif.

La loi de Planck [PM14], l'un des premiers résultats notables de la physique statistique, décrit alors la distribution spectrale de l'émission d'un corps noir :

$$I(\lambda,T) = \frac{2hc^2}{\lambda^5(exp(hc/\lambda k_B T)-1)}$$

où k_B représente la constante de Boltzmann. Le corps noir étant un émetteur diffusif, on peut en déduire l'énergie émise sur l'intégralité du demi-espace (on se place dans le cas le plus courant d'une surface solide) :

$$E(\lambda,T) = \pi I(\lambda,T) = \frac{2\pi hc^2}{\lambda^5(exp(hc/\lambda k_B T)-1)}$$

Cette fonction est représentée dans la figure 1.1.

La loi de Wien permet de déterminer le maximum de la distribution spectrale d'un corps noir pour une température donnée :

$$\frac{\partial E(\lambda,T)}{\partial \lambda} = 0 \Rightarrow \lambda T \approx 2900 \mu m.K$$

En intégrant la loi de Planck en fonction de la longueur d'onde, on obtient la loi de Stefan-Boltzmann qui représente l'intégralité de l'énergie émise par le corps noir à une température donnée.

$$E(T) = \int_0^\infty E(\lambda,T)d\lambda = \sigma T^4$$

où σ est la constante de Stefan-Boltzmann. L'échange entre 2 corps noirs aux températures T_a et T_b peut donc se mettre sous la forme :

$$q = \sigma(T_a^4 - T_b^4) \tag{1.4}$$

1.3 Résolution analytique

La plupart des problèmes thermiques nécessitent de connaître la température à l'intérieur d'un matériau. La conduction est donc le phénomène prépondérant dans ce type d'analyse. La convection et les radiations sont le plus souvent soit ignorées, soit intégrées comme des conditions limites. En considérant un milieu isotrope à caractéristiques thermophysiques constantes, l'équation de la chaleur (équation 1.2) peut alors se mettre sous la forme :

$$c_v \frac{\partial T(x,y,z,t)}{\partial t} = P(x,y,z,t) + div(k\overrightarrow{grad}T(x,y,z,t)) \tag{1.5}$$

où la fonction température $T(x,y,z,t)$ représente la température du système en un point et à un instant donnés, c_v la capacité thermique volumétrique, k la conductivité thermique et $P(x,y,z,t)$ la puissance produite.

Cette équation doit être complétée par des conditions limites qui peuvent être de 3 types :

- les conditions de première espèce, ou conditions de Dirichlet, qui représentent une température fixe ($T(x,y,z) = T_0$ où T_0 est une constante, cas notamment d'un changement d'état en surface du milieu),

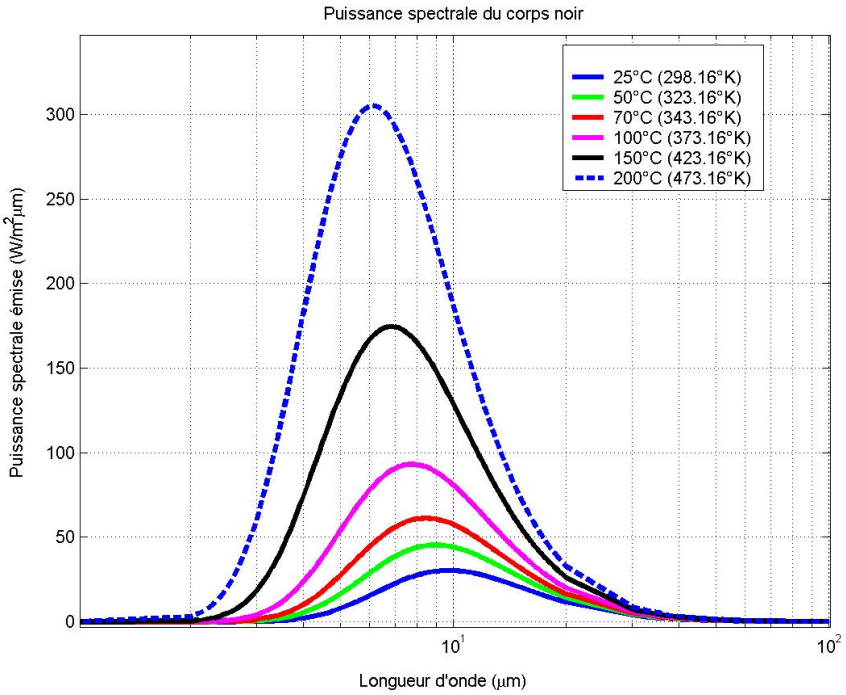


FIGURE 1.1 : Energie spectrale émise en fonction de la température pour un corps noir

- les conditions de deuxième espèce, ou conditions de Neumann, qui représentent un flux de chaleur fixé ($-k\frac{\partial T}{\partial n} = q_0$ où q_0 est une constante, cas des radiations (équation 1.4) quand la différence de températures entre 2 surfaces est importante),
- et les conditions de troisième espèce, ou conditions de Fourier, qui dépendent de la différence de température entre la surface et un autre milieu ($q = -k\frac{\partial T}{\partial n} = h_{e0}(T(x,y,z) - T_0)$ où h_{e0} est une constante représentant le coefficient d'échange et T_0 représente la température extérieure, cas de la convection (équation 1.3)).

où $\frac{\partial T}{\partial n}$ représente la dérivée normale à la surface.

De plus, lors de la résolution, il est également nécessaire de connaître les conditions initiales du système $T(x,y,z,0)$. Une fois ces informations réunies, on peut procéder à la résolution analytique du problème thermique.

Exemple 1 : Cas unidimensionnel en régime transitoire d'une plaque d'épaisseur finie L soumise à une température constante et uniforme.

La solution est de la forme $T(x,t)$. De plus, il n'y a pas de terme source. L'équation différentielle à résoudre s'écrit alors sous la forme :

$$c_v \frac{\partial T}{\partial t} - k\frac{\partial^2 T}{\partial^2 x} = 0$$

ou

$$\frac{\partial^2 T}{\partial^2 x} - \frac{1}{a}\frac{\partial T}{\partial t} = 0$$

où $a = \frac{k}{c_v}$ est appelée l'effusivité.

En incluant les conditions initiales (température supposée nulle à $t = 0$) et les conditions limites (une des limites est portée à une température constante), l'équation différentielle à résoudre est de la forme :

$$\begin{cases} \frac{\partial^2 T}{\partial^2 x} - \frac{1}{a}\frac{\partial T}{\partial t} = 0 \\ T(0,t) = T_{init}, T(L,t) = 0, T(x,0) = 0 \end{cases} \quad (1.6)$$

La première étape de la résolution consiste en la mise en forme du système d'équations. En effet, une des conditions limites n'est pas homogène. L'utilisation du principe de superposition va nous permettre de reporter cette non-homogénéité sur la condition initiale. On pose alors $T(x,t) = T_0(x) - T_1(x,t)$ où $T_0(x)$ représente le champ température final. On obtient ainsi 2 systèmes d'équations :

$$\begin{cases} \dfrac{\partial^2 T_1}{\partial^2 x} - \dfrac{1}{a}\dfrac{\partial T_1}{\partial t} = 0 \\ T_1(0,t) = 0,\ T_1(L,t) = 0,\ T_1(x,0) = T_0(x) \end{cases}$$

et

$$\begin{cases} \dfrac{\partial^2 T_0}{\partial^2 x} = 0 \\ T_0(0) = T_{init},\ T_0(L) = 0. \end{cases}$$

La solution pour T_0 est simple : $T_0 = T_{init}\left(1 - \dfrac{x}{L}\right)$ et elle permet d'obtenir la condition initiale utilisée pour trouver T_1.

La deuxième étape de la résolution repose sur la séparation des variables. La fonction $T_1(x,t)$ est alors mise sous la forme $T_1(x,t) = u(x)v(t)$. En introduisant ces fonctions dans l'équation différentielle, on obtient la relation suivante :

$$\frac{u''(x)}{u(x)} = \frac{1}{a}\frac{v'(t)}{v(t)} = cste$$

En supposant la constante négative, on peut alors écrire $cste = -\alpha^2$. Avec les conditions limites, les k-ième solutions particulières de chacune des équations différentielles prennent la forme :

$$\begin{cases} u(x) = Asin(\alpha_k x) \\ v(t) = Cexp(-a\alpha_k^2 t) \end{cases} \quad \text{avec } \alpha_k = \frac{k\pi}{L} \text{ et } k \in \mathbb{N}$$

La solution générale, compte tenu de la linéarité des équations, peut alors être mise sous la forme d'une combinaison linéaire des solutions particulières :

$$T_1(x,t) = \sum_k E_k sin(\alpha_k x) exp(-a\alpha_k^2 t)$$

où E_k représente le coefficient de pondération de la solution particulière k dans la solution finale.

La condition initiale peut alors s'écrire :

$$T_1(x,0) = T_0(x) = \sum_k E_k sin\,(\alpha_k x)$$

Cette équation va nous permettre de calculer les E_k. En effet, les fonctions $f(x) = sin\,(\alpha_k x)$ et $g(x) = sin\,(\alpha_{k'} x)$ sont orthogonales si $k \neq k'$ pour le produit scalaire défini par $< f(x), g(x) >= \displaystyle\int_0^L f(x)g(x)dx$. Pour déterminer les E_k, on peut alors écrire :

$$\int_0^L T_0(x) sin(\alpha_k x)dx = \int_0^L \sum_{k'} E_{k'} sin(\alpha_{k'} x) sin(\alpha_k x)dx = E_k \int_0^L sin^2(\alpha_k x) = \frac{L}{2}E_k$$

$$\text{soit } E_k = \frac{2}{L}\int_0^L T_0(x) sin(\alpha_k x)dx$$

En insérant cette nouvelle expression de E_k dans l'équation générale, on obtient :

$$T_1(x,t) = \sum_k sin(\alpha_k x)\left(\frac{2}{L}\int_0^L T_0(x') sin(\alpha_k x')dx'\right) exp(-a\alpha_k^2 t)$$

On peut ensuite utiliser l'expression de $T_0(x)$ pour obtenir :

$$T_1(x,t) = \frac{2T_{init}}{\pi} \sum_k \frac{sin(\alpha_k x)}{k} exp(-a\alpha_k^2 t)$$

Pour le problème originel, on a alors :

$$T(x,t) = T_0(x) - T_1(x,t) = T_{init}\left(1 - \frac{x}{L}\right) - \frac{2T_{init}}{\pi} \sum_k \frac{sin(\alpha_k x)}{k} exp(-a\alpha_k^2 t)$$

Cette démarche permet de trouver la solution exacte du problème de prédiction de température en tout point du système et à tout instant. Cependant, si ce type d'analyse est possible dans un cas relativement simple, il s'avère très difficile voire impossible d'utiliser cette méthode de résolution lorsque l'architecture du modèle étudiée devient trop complexe ou lorsqu'il est impossible d'homogénéiser les conditions limites. Toutefois, dans des conditions particulières, il est possible de trouver une solution pour des architectures complexes. Un exemple de ce type d'approche peut être trouvé dans [Muz06].

1.4 Résolution numérique

Résoudre parfaitement le système d'équations différentielles et de conditions limites qui découle de l'architecture du système modélisé est impossible dans la plupart des cas réels. La seule voie restante est de s'en remettre à des solutions approchées, en essayant de réduire le plus possible l'erreur liée à l'approximation. Les méthodes numériques offrent ainsi une alternative avantageuse lorsque le modèle devient trop complexe. La solution n'est plus alors le champ complet de température en tout point de l'espace et du temps. Elle prend la forme d'un vecteur représentant la température en chaque noeud d'un maillage. Ce vecteur évolue alors itérativement entre chaque pas de temps par l'intermédiaire d'opérations matricielles. Les différentes méthodes de résolution numérique sont présentées dans de nombreux ouvrages et notamment celui de Goncalvès [Gon05]. Les exemples de ce chapitre sont adaptés de ceux de cet ouvrage.

1.4.1 Différences finies

La méthode des différences finies consiste à tronquer le développement de Taylor d'une fonction afin de remplacer les dérivées partielles de l'équation différentielle. Ainsi, dans le cas de l'équation de la chaleur, on considère toujours la fonction $T(x,y,z,t)$ représentant la température en tout point du système en fonction du temps. Soit Δx un déplacement infinitésimal, si on étudie le développement de cette fonction température au voisinage de x, on peut écrire :

$$T(x+\Delta x,y,z,t) = T(x,y,z,t) + \Delta x \frac{\partial T}{\partial x} + \frac{\Delta x^2}{2}\frac{\partial^2 T}{\partial x^2} + \frac{\Delta x^3}{6}\frac{\partial^3 T}{\partial x^3} + \ldots + \frac{\Delta x^i}{i!}\frac{\partial^i T}{\partial x^i} + \ldots$$

Si on tronque ce développement au premier ordre, on obtient :

$$\frac{\partial T}{\partial x} + \mathcal{O}(\Delta x) = \frac{T(x+\Delta x,y,z,t) - T(x,y,z,t)}{\Delta x}$$

En étendant le même raisonnement aux autres dimensions spatiales, il est possible d'obtenir un système linéaire à partir de l'équation différentielle. Pour utiliser cette méthode de résolution, on constate qu'un maillage hexaèdrique doit être utilisé ce qui peut limiter la géométrie des systèmes considérés.

Exemple 2 : Cas unidimensionnel en régime transitoire d'une plaque d'épaisseur finie L soumise à une température constante et uniforme.

Comme dans l'exemple 1, l'équation différentielle à résoudre est de la forme :

$$\begin{cases} \dfrac{\partial^2 T}{\partial^2 x} - \dfrac{1}{a}\dfrac{\partial T}{\partial t} = 0 \\ T(0,t) = T_{init},\ T(L,t) = 0,\ T(x,0) = 0 \end{cases}$$

On considère les déplacements infinitésimaux Δx dans l'espace et Δt dans le temps. On note i l'indice des mailles le long de l'axe x et n l'indice du pas de temps considéré. Ainsi, on a $x_i = i\Delta x$ et $t_n = n\Delta t$ et on notera par la suite $T_n^i = T(x_i, t_n)$. La plaque étant d'épaisseur finie, on considère de plus l le nombre de mailles, tel que $1 \leq i \leq l$.

On peut alors écrire :

$$\left(\frac{\partial^2 T}{\partial^2 x}\right)_n^i - \frac{1}{a}\left(\frac{\partial T}{\partial t}\right)_n^i = 0$$

Chacune des dérivées partielles peut alors être traduite sous forme discrète :

$$\left(\frac{\partial^2 T}{\partial^2 x}\right)_n^i = \frac{T_n^{i+1} - 2T_n^i + T_n^{i-1}}{\Delta x^2}$$
$$\left(\frac{\partial T}{\partial t}\right)_n^i = \frac{T_{n+1}^i - T_n^i}{\Delta t}$$

En posant $\delta = a\dfrac{\Delta t}{\Delta x^2}$, on obtient l'équation :

$$T_{n+1}^i = \delta T_n^{i-1} + (1-2\delta)T_n^i + \delta T_n^{i+1}$$

En utilisant une notation matricielle et les conditions limites :

$$\begin{bmatrix} T^1 \\ T^2 \\ T^3 \\ \vdots \\ T^{l-3} \\ T^{l-2} \\ T^{l-1} \end{bmatrix}_{n+1} = \begin{bmatrix} 1-2\delta & \delta & 0 & \dots & 0 & 0 & 0 \\ \delta & 1-2\delta & \delta & 0 & \dots & 0 & 0 \\ 0 & \delta & 1-2\delta & \delta & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & \delta & 1-2\delta & \delta & 0 \\ 0 & 0 & \dots & 0 & \delta & 1-2\delta & \delta \\ 0 & 0 & 0 & \dots & 0 & \delta & 1-2\delta \end{bmatrix} \begin{bmatrix} T^1 \\ T^2 \\ T^3 \\ \vdots \\ T^{l-3} \\ T^{l-2} \\ T^{l-1} \end{bmatrix}_n + \delta \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ T_{init} \end{bmatrix} \tag{1.7}$$

La température en chaque point du maillage se déduit donc pour chaque itération de ce système linéaire.

1.4.2 Volumes finis

Le principe des volumes finis consiste à intégrer l'équation différentielle considérée sur chaque volume élémentaire défini par le maillage. Pour l'équation de la chaleur, on considère toujours la fonction température $T(x,y,z,t)$. On repart de l'équation différentielle 1.2 :

$$c_v \frac{\partial T}{\partial t} = P + div(k\overrightarrow{grad}T)$$

ou

$$\frac{1}{a}\frac{\partial T}{\partial t} - div(\overrightarrow{grad}T) = P$$

En intégrant sur le volume Vd'une maille, on obtient :

$$\frac{1}{a}\frac{\partial}{\partial t}\int_V T dV - \int_V div(\overrightarrow{grad}T)dV = \int_V P dV$$

On note S la surface de la maille considérée et n la normale à cette surface. À partir du théorème d'Ostrogradski, on peut écrire :

$$\frac{1}{a}\frac{\partial}{\partial t}\int_V T dV - \oint_S \overrightarrow{grad}T d\vec{S} = \int_V P dV$$

Pour le premier terme de l'équation, si la température T est supposée constante au sein de la maille, on obtient :

$$\frac{\partial}{\partial t}\int_V T dV = V\left(\frac{\partial T}{\partial t}\right)_{maille}$$

De plus, en discrétisant le domaine temporel et en posant ΔT la variation de température entre 2 instants, ce terme peut se mettre sous la forme :

$$V\left(\frac{\partial T}{\partial t}\right)_{maille} = V\frac{\Delta T}{\Delta t}$$

Pour le second terme de l'équation, si on suppose le flux $\overrightarrow{grad}T$ constant sur chaque face de la maille, on peut écrire :

$$\oint_S \overrightarrow{grad}T d\vec{S} = \sum_{S_{maille}} \left(\overrightarrow{grad}T\right)_{maille} S_{maille}\vec{n}_{maille}$$

où S_{maille} est la surface de la face de la maille considérée et $\vec{n}_{maille}$ le vecteur unitaire normal à cette face orienté vers l'extérieur.

Si, de plus, la puissance générée au sein de la maille est également considérée comme constante, l'équation globale discrétisée est alors :

$$\frac{1}{a}V\frac{\Delta T}{\Delta t} - \sum_{S_{maille}} \left(\overrightarrow{grad}T\right)_{maille} S_{maille}\vec{n}_{maille} = PV$$

Nous considérons l'application de ce principe sur l'exemple

Exemple 3 : Cas unidimensionnel en régime transitoire d'une plaque d'épaisseur finie L soumise à une température constante et uniforme.

On prend des conventions identiques à celles de l'exemple 2. Toutefois, dans le cas des volumes finis, x_i représente le centre de la maille. En intégrant sur le volume d'une maille, on obtient alors :

$$\int_{x_{i-1/2}}^{x_{i+1/2}} \frac{\partial^2 T}{\partial^2 x} - \frac{1}{a}\int_{x_{i-1/2}}^{x_{i+1/2}} \frac{\partial T}{\partial t} = 0$$

Dans notre cas, on a $V = \Delta x$ et $P = 0$. En utilisant une discrétisation de type volume fini, on peut mettre l'équation sous la forme :

$$\frac{1}{a}\left(\frac{T_{n+1}^i - T_n^i}{\Delta t}\right)\Delta x = \left[\left(\frac{\partial T}{\partial x}\right)_n^{x_{i+1/2}} - \left(\frac{\partial T}{\partial x}\right)_n^{x_{i-1/2}}\right]$$

On calcule la valeur moyenne du flux à la surface de la maille :

$$\left(\frac{\partial T}{\partial x}\right)_n^{x_{i+1/2}} = \frac{1}{\Delta x}\int_{x_i}^{x_{i+1}} \frac{\partial T}{\partial x} dx = \frac{T_n^{i+1} - T_n^i}{\Delta x}$$

Pour les mailles situées aux limites, on doit adapter cette équation et on obtient :

$$\left(\frac{\partial T}{\partial x}\right)_n^{x_{l+1/2}} = -2\frac{T_n^l}{\Delta x},\ \left(\frac{\partial T}{\partial x}\right)_n^{x_{1/2}} = 2\frac{T_n^1 - T_{init}}{\Delta x}$$

En posant $\delta = a\dfrac{\Delta t}{\Delta x^2}$, on obtient l'équation :

$$\begin{aligned} T_{n+1}^i &= \delta T_n^{i-1} + (1-2\delta)T_n^i + \delta T_n^{i+1} \\ T_{n+1}^l &= (1-3\delta)T_n^l + \delta T_n^{l-1} \\ T_{n+1}^1 &= \delta T_n^2 + (1-3\delta)T_n^1 + 2\delta T_{init} \end{aligned}$$

Soit, sous une notation matricielle :

$$\begin{bmatrix} T^1 \\ T^2 \\ T^3 \\ \vdots \\ T^{l-3} \\ T^{l-2} \\ T^{l-1} \end{bmatrix}_{n+1} = \begin{bmatrix} 1-3\delta & \delta & 0 & \dots & 0 & 0 & 0 \\ \delta & 1-2\delta & \delta & 0 & \dots & 0 & 0 \\ 0 & \delta & 1-2\delta & \delta & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & \delta & 1-2\delta & \delta & 0 \\ 0 & 0 & \dots & 0 & \delta & 1-2\delta & \delta \\ 0 & 0 & 0 & \dots & 0 & \delta & 1-3\delta \end{bmatrix} \begin{bmatrix} T^1 \\ T^2 \\ T^3 \\ \vdots \\ T^{l-3} \\ T^{l-2} \\ T^{l-1} \end{bmatrix}_n + 2\delta \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ T_{init} \end{bmatrix} \tag{1.8}$$

1.4.3 Éléments finis

La méthode des éléments finis consiste à résoudre l'équation différentielle dans un sous-espace de fonctions de dimension finie plutôt que dans l'espace des fonctions continues de dimension infinie. Ce sous-espace est défini par un ensemble de fonctions élémentaires linéairement indépendantes, notées ϕ_i. Ces fonctions sont elles-mêmes le plus souvent définies comme des fonctions polynomiales sur les noeuds d'un maillage de l'espace. La solution est alors obtenue sous la forme d'une somme pondérée de ces fonctions élémentaires. Une illustration de cette démarche est donnée dans l'exemple suivant

Exemple 4 : Cas unidimensionnel en régime transitoire d'une plaque d'épaisseur finie L soumise à une température constante et uniforme.

On prend des conventions identiques à celles de l'exemple 2. On choisit des fonctions élémentaires polynomiales de degré 1 :

$$\phi_i(x) = \begin{cases} \dfrac{x - x_{i-1}}{x_i - x_{i-1}} \text{ si } x_{i-1} < x < x_i \\ \\ \dfrac{x - x_{i+1}}{x_i - x_{i+1}} \text{ si } x_i < x < x_{i+1} \\ \\ 0 \text{ sinon.} \end{cases}$$

Le nombre de fonctions élémentaires est alors égal au nombre de mailles dans les exemples précédents ($1 \leq i \leq l$).

On cherche une solution de la forme :

$$T(x,t) = f(x)g(t) = \left(\sum_i g_i(t)\phi_i(x)\right)$$

En utilisant cette expression dans l'équation différentielle, on obtient :

$$\left(\sum_i g_i(t)\frac{d^2\phi_i(x)}{dx^2}\right) - \frac{1}{a}\left(\sum_i \frac{dg_i(t)}{dt}\phi_i(x)\right) = 0$$

Pour toute fonction $v(x)$, on peut alors écrire :

$$\int\left(\sum_i g_i(t)\frac{d^2\phi_i(x)}{dx^2}\right)v(x)dx - \frac{1}{a}\int\left(\sum_i \frac{dg_i(t)}{dt}\phi_i(x)\right)v(x)dx = 0$$

Si de plus $v(x)$ est nulle en dehors du domaine $[0,L]$, alors en intégrant par partie, on a :

$$\int_0^L\left(\sum_i g_i(t)\frac{d\phi_i(x)}{dx}\right)\frac{dv(x)}{dx}dx + \frac{1}{a}\int_0^L\left(\sum_i \frac{dg_i(t)}{dt}\phi_i(x)\right)v(x)dx = 0$$

En particulier, si on choisit $v(x) = \phi_j(x)$, il suit :

$$\sum_i g_i(t)\left(\int_0^L \frac{d\phi_i(x)}{dx}\frac{d\phi_j(x)}{dx}dx\right) + \frac{1}{a}\sum_i \frac{dg_i(t)}{dt}\left(\int_0^L \phi_i(x)\phi_j(x)dx\right) = 0$$

Si on décrit les matrices $\mathbf{M}$ et $\mathbf{K}$ et le vecteur $G(t)$ tels que :

$$\begin{aligned}
\mathbf{M}_{ij} &= \left(\int_0^L \phi_i(x)\phi_j(x)dx\right)\\
\mathbf{K}_{ij} &= \left(\int_0^L \frac{d\phi_i(x)}{dx}\frac{d\phi_j(x)}{dx}dx\right)\\
G_i(t) &= [g_1(t)g_2(t)g_3(t)\dots g_i(t)\dots g_l(t)]^\top
\end{aligned}$$

Compte tenu de la définition des fonctions élémentaires, les matrices $\mathbf{M}$ et $\mathbf{K}$ sont tridiagonales. On obtient alors avec les mêmes conventions que dans les exemples précédents pour les déplacements infinitésimaux Δx et Δt et les indices i et n :

$$\mathbf{K}_{ij} = \begin{cases} \dfrac{2}{\Delta x} \text{ si } i = j \\ -\dfrac{1}{\Delta x} \text{ si } i - j = \pm 1 \\ 0 \text{ sinon.} \end{cases}$$

$$\mathbf{M}_{ij} = \begin{cases} \dfrac{2\Delta x}{3} \text{ si } i = j \\ -\dfrac{\Delta x}{6} \text{ si } i - j = \pm 1 \\ 0 \text{ sinon.} \end{cases}$$

Alors, on peut alors écrire :

$$\frac{1}{a}\mathbf{M}\frac{dG(t)}{dt} + \mathbf{K}G(t) = 0$$

En utilisant une intégration temporelle de type Euler et en posant $\delta = a\dfrac{\Delta t}{\Delta x^2}$, on peut écrire cette équation sous la même forme que pour les schémas de discrétisation précédents (équations 1.7 et 1.8).

On constate donc que la méthodes des différences finies et celles des volumes finies sont relativement proches pour notre exemple. Toutefois, la méthode de volumes finis utilise, comme la méthode des éléments finis, des approximations d'intégrales, alors que la méthode des différences finies repose sur des approximations de dérivées. De plus, la méthode des volumes finis et celle des éléments finis apportent plus de souplesse quant au choix du

maillage possible. C'est pourquoi la méthode des différences finies n'est utilisé que pour des cas très simples. Grâce à la possibilité de choisir les fonctions élémentaires, la méthode des éléments finis est souvent plus précise que celle des volumes finis mais la complexité algorithmique est plus élevée. Le choix entre ces deux méthodes revient donc à un compromis entre la précision attendue et la complexité des calculs nécessaires.

1.5 Réduction d'ordre des modèles

La taille des systèmes linéaires générés dépend du nombre de mailles utilisées. Or, la taille des mailles doit être adaptée à celle des objets modélisés. Si l'on veut modéliser avec finesse un ensemble d'objets de différentes dimensions, on obtient ainsi le plus souvent des systèmes linéaires de très grande taille.

Les modèles thermiques s'insèrent dans une famille plus large de modèles, dits modèles d'états. En effet, si l'on considère les vecteurs $T(t) \in \mathbb{R}^m$ et $P(t) \in \mathbb{R}^p$ représentant la température sur chaque noeud du maillage à l'instant t et la puissance injectée dans le système, où m désigne le nombre de mailles du modèle et p le nombre de fonctions servant à décrire la puissance injectée, alors les modèles numériques obtenus peuvent se mettre sous la forme, avec $\mathbf{A} \in \mathbb{R}^{m\times m}$ et $\mathbf{B} \in \mathbb{R}^{m\times p}$:

$$\frac{dT(t)}{dt} = \mathbf{A}\,T(t) + \mathbf{B}\,P(t)$$

Cette forme correspond parfaitement à la définition d'un modèle d'état (ces modèles seront décrits plus précisément dans le chapitre 4). Dans la suite, les états seront notés $\mathbf{x}(t)$, les entrées du modèle $\mathbf{u}(t)$ et les sorties du modèle $\mathbf{y}(t)$. On considère donc la représentation d'état suivante [1] :

$$\frac{d\mathbf{x}(t)}{dt} = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t)$$

ou

$$\mathbf{x}_{n+1} = \mathbf{A}\mathbf{x}_n + \mathbf{B}\mathbf{u}_n$$

dans le cas discret.

Le plus souvent, seule la température en quelques positions particulières est réellement utile. Le système peut alors se mettre sous la forme plus générale, où la matrice $\mathbf{C}$ sert en thermique à sélectionner la température des mailles qui nous intéressent :

$$\begin{cases} \mathbf{x}_{n+1} = \mathbf{A}\mathbf{x}_n + \mathbf{B}\mathbf{u}_n \\ \mathbf{y}_n = \mathbf{C}\mathbf{x}_n \end{cases}$$

Même dans un espace tridimensionnel, les matrices caractérisant les modèles thermiques sont relativement creuses. Les méthodes de réduction de modèle sont parfaitement adaptées à ce type de problématique. De plus, le nombre de sorties $\mathbf{y}(t)$ et d'entrées $\mathbf{u}(t)$ du modèle est fréquemment très inférieur à la dimension de $\mathbf{x}(t)$. Enfin, physiquement, deux mailles proches vont nécessairement présenter une corrélation très forte entre leurs températures. Ces faits renforcent les prédispositions du modèle pour les méthodes de réduction. L'objectif est donc de trouver un autre système linéaire de taille très inférieure et capable de produire une bonne approximation des sorties $\mathbf{y}(t)$. Les équations étant linéaires, on choisit une projection matricielle du type $\mathbf{x}(t) = \mathbf{V}\mathbf{w}(t)$ où $\mathbf{w}(t)$ sont les états du nouveau système et $\mathbf{V}$ représente une matrice de projection entre les espaces d'états des deux systèmes. Le nouveau système peut alors se mettre sous la forme :

$$\begin{cases} \frac{d\mathbf{w}(t)}{dt} = \mathbf{V}^{-1}\mathbf{A}\mathbf{V}\mathbf{w}(t) + \mathbf{V}^{-1}\mathbf{B}\mathbf{u}(t) \\ \mathbf{y}(t) = \mathbf{C}\mathbf{V}\mathbf{w}(t) \end{cases}$$

1. Dans le cas le plus général, l'équation prend la forme de $\mathbf{E}\frac{d\mathbf{x}(t)}{dt} = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t)$ mais pour la modélisation thermique, la matrice $\mathbf{E}$ est toujours inversible.

La détermination de la matrice $\mathbf{V}$ constitue l'essentiel du problème. Un grand nombre d'algorithmes ont été proposés pour calculer de manière efficace cette matrice et obtenir la meilleure précision sur l'estimation de la sortie pour un nombre d'états réduit. Ces algorithmes peuvent sommairement être classés en deux catégories :

- celles fondées sur des méthodes de projection et qui font une utilisation intensive de la décomposition en valeurs singulières (Singular perturbation method, Proper Orthogonal Decomposition (POD), ...),
- et celles qui reposent sur le « moment matching » (algorithmes de Lanczos et d'Arnoldi, Rational Krylov Method, ...).

Les méthodes de projection vont chercher à projeter les états du système d'origine vers une nouvelle base où les états faiblement observables (qui n'ont qu'une faible influence sur la sortie du système) vont être éliminés. Pour le « moment matching », il s'agit de trouver une fonction de transfert dont un certain nombre de pôles correspondent à ceux du système réel. Une revue presque exhaustive de ces algorithmes pourra être trouvée par exemple dans l'article d'Antoulas et al. [ASG01]. Les méthodes à base de projection offrent l'avantage de garantir une borne sur l'erreur commise par le modèle réduit et préservent la stabilité du modèle. Les méthodes de « moment matching » sont cependant plus légères algorithmiquement et peuvent donc traiter des systèmes de plus grandes tailles. Toutes ces méthodes permettent dans la plupart des cas de réduire fortement le nombre de mailles du modèle tout en conservant une précision acceptable.

Exemple 5 : Réduction de modèle thermique instationnaire en 2D.

On considère un système simple, constitué par un disque qui reçoit des impulsions de puissance et qui est situé au centre d'une plaque carrée dont les bords sont maintenus à une température constante. La géométrie ainsi que le maillage utilisé sur le système sont présentés sur la figure 1.2. Le maillage est constitué de 725 noeuds, ce qui correspond également à l'ordre du modèle créé.

À partir du maillage et des caractéristiques thermiques des différents matériaux, on peut effectuer une discrétisation par éléments finis et établir une équation de la forme :

$$\mathbf{M}\frac{dT(t)}{dt} + \mathbf{K}T(t) = F(t)$$

On ne s'intéresse qu'à la réponse d'un unique noeud, celui dont la température est la plus élevée. L'impulsion de puissance appliquée sur le disque central ainsi que la réponse de ce noeud sont présentées à la figure 1.3. La méthode de réduction utilisée est assez simple et se rapproche d'une POD, même si cette méthode est généralement réservée au cas non-linéaire. On part d'une séquence d'impulsions injectées dans le modèle pour générer une séquence d'états, qui correspondent ici aux températures sur les différents noeuds : $\mathbf{T}_t = [T(1)\,T(2)\,\ldots\,T(d-1)\,T(d)]^\top \in \mathbb{R}^{d\times m}$. Pour déterminer les états du modèles réduits, on applique une décomposition en valeurs singulières (méthode décrite dans l'annexe B) sur la matrice $\mathbf{T}_t$:

$$\mathbf{T}_t = \mathbf{U}_r\mathbf{S}_r\mathbf{V}_r^T$$

Par convention, on suppose que les valeurs singulières placées sur la diagonale de $\mathbf{S}_r$ sont triées par ordre décroissant. Pour obtenir un système réduit donnant une bonne approximation, on peut alors choisir la matrice de projection $\mathbf{V} = \mathbf{U}_r(1:n,:)$ où n représente l'ordre sélectionné pour le modèle. Les résultats obtenus pour différentes valeurs de n sont présentés sur la figure 1.4. On constate que plus l'ordre du modèle réduit augmente, plus la réponse fournie par ce modèle se rapproche de la vraie réponse du système thermique. En particulier pour un ordre $n = 10$, la superposition de la réponse correspondante avec la vraie température est parfaite. Il est à noter qu'un ordre $n = 5$ assure déjà une approximation quasi parfaite.

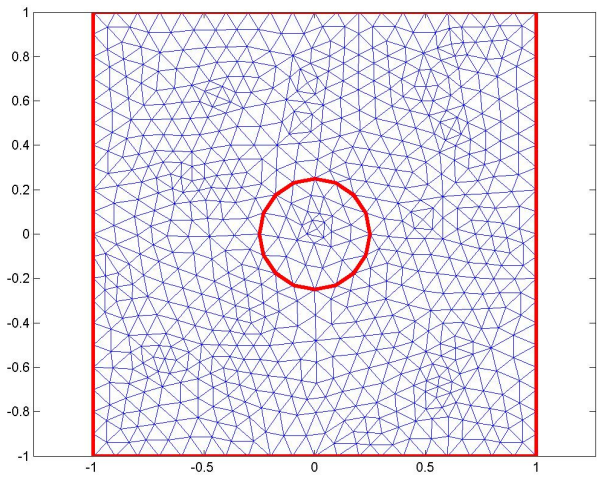


FIGURE 1.2 : Géométrie et maillage pour l'exemple de réduction de modèle.

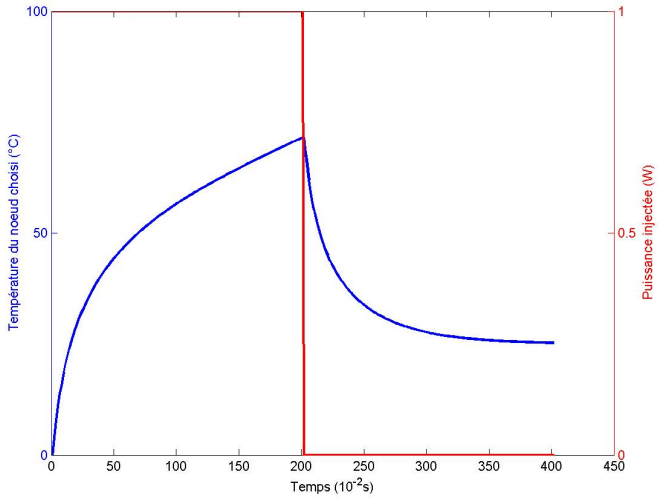


FIGURE 1.3 : Impulsion de puissance injectée et réponse en température du noeud choisi pour l'exemple de réduction de modèle. En rouge l'impulsion et en bleu la réponse obtenue.

1.6 Applications

De nombreux logiciels implémentent les méthodes de résolution numérique pour modéliser différents problèmes thermiques. Le système est alors représenté sous la forme de plusieurs blocs de matériaux homogènes mis bout à bout pour obtenir une structure réaliste. Cette approche est utilisée sur un problème-jouet où un composant (de type transistor de puissance) est monté sur un système de refroidissement (typiquement un système d'ailettes). Il est à noter que ce cas de figure se rapproche de la problématique envisagée dans cette thèse à savoir la modélisation de l'évolution de la température interne des composants radar dans un cas simplifié. Compte tenu des dimensions très différentes de ces 2 éléments, ils doivent être étudiés de manière séparée pour conserver des temps de calcul raisonnables. En effet, la précision de l'approximation dépend grandement de la finesse du maillage choisi.

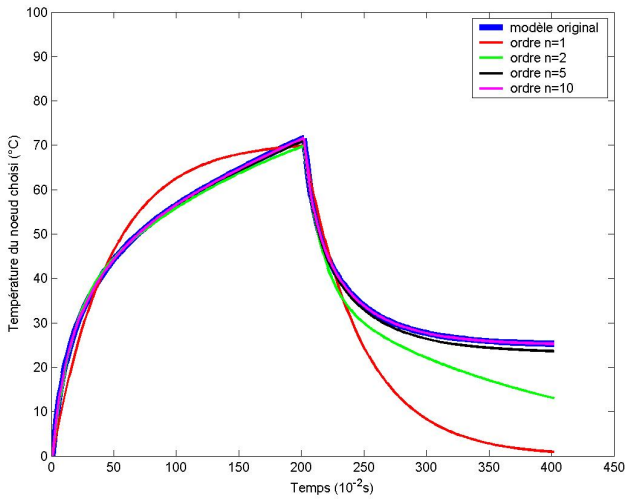


FIGURE 1.4 : Réponse des modèles réduits pour différentes valeurs de l'ordre n du modèle réduit.

1.6.1 Modèle complet simplifié

Un premier modèle a été développé sous le logiciel TAS [rdpA]. Ce logiciel utilise les éléments finis et autorise donc des mailles de différentes formes. Ce modèle reprend l'architecture complète du problème-jouet, avec un composant électronique placé sur une plaque munie d'ailettes. On suppose un écoulement d'air constant à $10m.s^{-1}$ dans les ailettes. Seuls les échanges convectifs les plus importants sont modélisés, ce qui permet de les représenter par un simple coefficient d'échange associé à la surface interne des ailettes. Les échanges radiatifs n'ont pas été pris en compte car ils sont négligeables dans ces conditions. L'architecture du composant a été grandement simplifiée pour permettre de réaliser le modèle (figure 1.5). La partie active du composant est soumise à plusieurs trains d'impulsions de puissance, répartis en deux phases d'activité, chacune suivie d'une phase de repos. La température de jonction relevée est associée à la plus forte température observée dans le modèle.

Le signal relevé montre la superposition de 2 dynamiques dans la réponse thermique du système (figure 1.6). La dynamique la plus lente est liée au radiateur à ailettes. Compte tenu de sa masse et de ses dimensions, il possède une forte inertie thermique. Durant les phases d'activité, l'amplitude de cette dynamique est de l'ordre de 10°C pour un temps de réponse supérieur à la seconde. La dynamique la plus rapide est liée à la réponse thermique du composant. Toutefois, étant donné que l'architecture du composant a été simplifiée, l'amplitude de cette dynamique a été minorée et son temps de réponse paraît plus important. On constate une amplitude d'environ 20°C pour un temps de réponse de quelques milisecondes.

Ce modèle permet principalement de mettre en évidence la difficulté de construire un modèle qui puisse intégrer l'intégralité du problème-jouet. En effet, outre les complications liées aux différentes échelles à intégrer dans le modèle, la génération de bases de données pertinentes pose problème. La dynamique du système de refroidissement nécessite un temps d'acquisition long et celle du composant ne peut être observée qu'avec un taux d'échantillonnage important. La base de données nécessaire pour observer les deux dynamiques doit donc être très grande. De plus, avec une fréquence d'échantillonnage élevée, l'influence de la dynamique lente est très difficilement observable sur un faible nombre d'échantillons, ce qui pose problème lors de l'application d'algorithmes d'apprentissage.

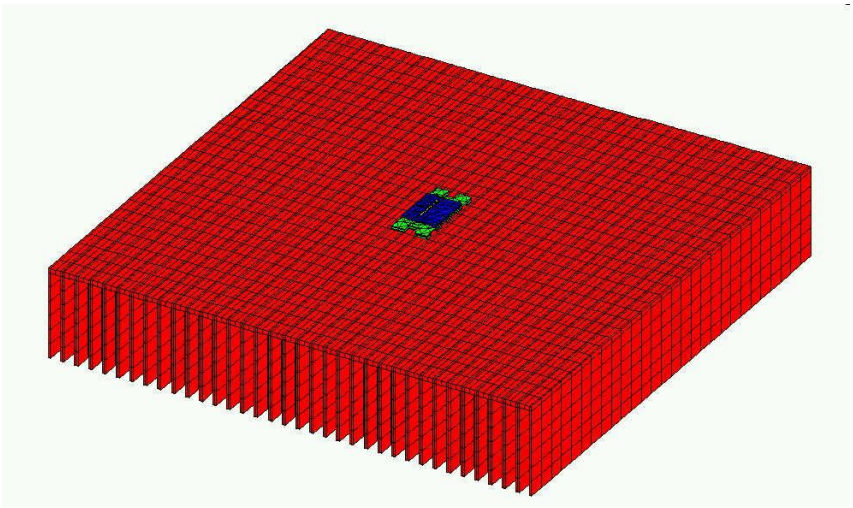


FIGURE 1.5 : Modèle complet du problème-jouet réalisé sous le logiciel TAS.

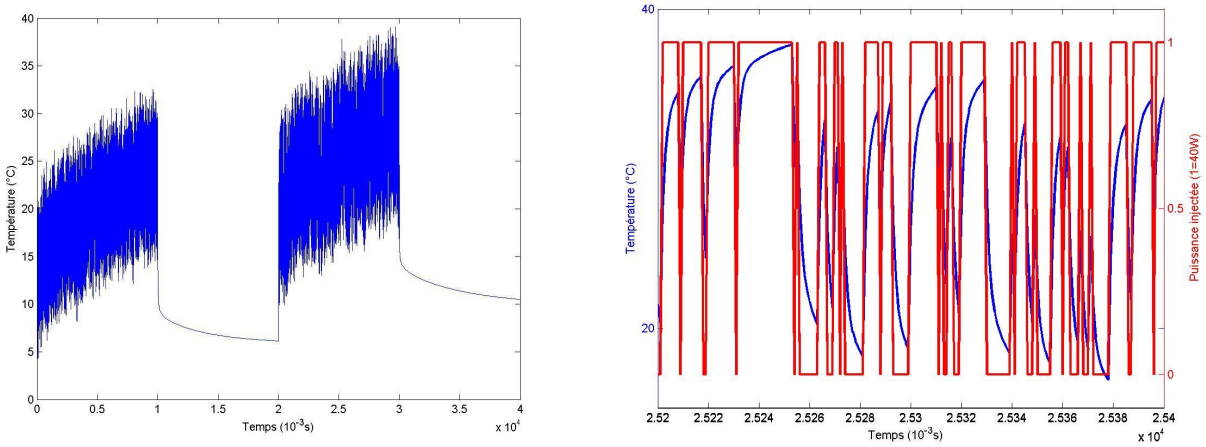


FIGURE 1.6 : Sortie obtenue pour le modèle réalisé sous TAS. À gauche, l'intégralité du signal obtenu où l'on peut deviner la dynamique lente liée au système de refroidissement. À droite, un zoom sur les données pour observer la dynamique rapide liée à la réponse thermique du composant avec les impulsions de puissance injectées.

1.6.2 Modèle du composant détaillé

Un second modèle a été réalisé sous le logiciel Flotherm [Flo]. Utilisant les volumes finis, Flotherm offre une importante bibliothèque d'objets et de matériaux et permet des simulations plus complètes que TAS. Le modèle ne représente que le transitor mais intègre l'intégralité de son architecture (figure 1.7). Compte tenu des symétries que présente son architecture, seul un quart de transistor est modélisé. Le modèle s'arrête à la base du transistor, qui est supposée maintenue à une température fixée de 40˚C. Ce modèle néglige donc les dynamiques les plus lentes liées à la dérive thermique du système de refroidissement qui se trouve normalement sous le composant. Les transferts radiatifs et convectifs ont été négligés compte tenu de leurs faibles influences pour ce cas.

Deux cas ont été étudiés pour ce modèle. En effet, la conductivité thermique du semi-conducteur utilisé pour le transistor, l'arséniure de gallium(AsGa), présente une dépendance vis-à-vis de la température. Si cette dépendance est prise en compte, le modèle devient non linéaire par définition (puisque la matrice de conductance va dépendre de la température en chacun des noeuds). Afin d'analyser l'importance de cette non-linéarité, la réponse du modèle avec ou sans la prise en compte de la dépendance thermique de la conductivité de l'AsGa a été relevée. Le modèle a été soumis à des impulsions de puissance. La zone active a été localisée sur les doigts du transistor et l'évolution temporelle de la température a été relevée à cette position (figure 1.8).

Ces courbes nous permettent de voir que l'influence de la non-linéarité est assez faible (10˚C au maximum). De plus, avec la faible inertie du modèle, la dérive entre les 2 courbes reste limitée. Cependant, la différence entre les 2 cas est la plus forte au sommet des pics de températures, et c'est ce paramètre qui est le plus essentiel pour cette étude. On considère en effet qu'au-delà de 180˚C, il y a dégradation de la fiabilité des composants. Pour la suite, cette non-linéarité sera donc conservée. Enfin, ce modèle permet de pallier les manques du modèle précédent et nous offre une base de données sur les dynamiques les plus rapides du système, avec des élévations de température réalistes de l'ordre de 100˚C en quelques dizaines de microsecondes.

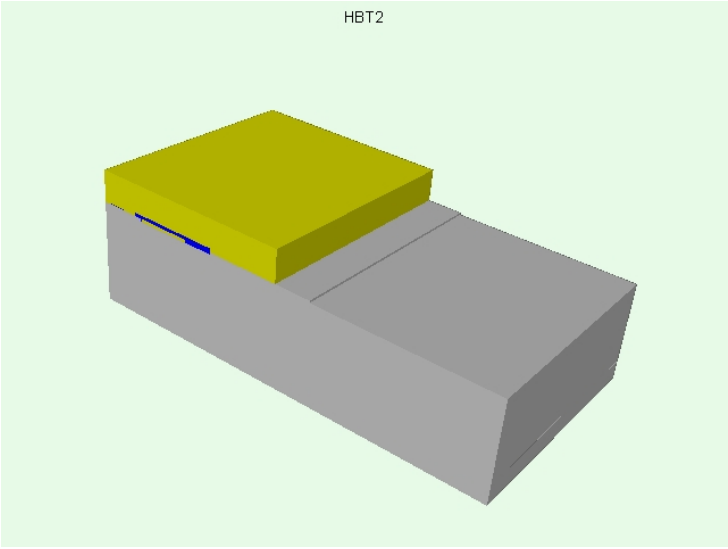


FIGURE 1.7 : Modèle de transistor réalisé sous le logiciel Flotherm.

1.7 Conclusion

L'approche physique permet une compréhension fine des phénomènes mise en jeu. De plus, même si les modèles générés se révèlent très grands, ils sont également très flexibles et donnent accès à la température en chaque maille du modèle. Toutes ces informations ne sont pas utiles pour notre étude mais les données générées nous ont ainsi permis de mieux comprendre les problématiques liées à la modélisation thermique et de fixer les ordres de

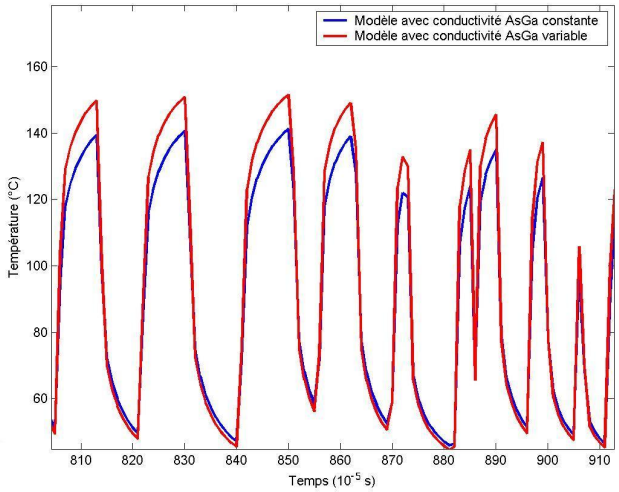


FIGURE 1.8 : Sorties obtenues pour le modèles de transistor réalisés sous le logiciel Flotherm. En bleu, la sortie du modèle où la dépendance de la conductivité de l'AsGa vis-à-vis de la température n'a pas été prise en compte. En rouge, celle du modèle où cette dépendance a été prise en compte.

grandeurs pour les amplitudes et les temps de réponse des signaux de température. Tout d'abord, les données du premier modèle ont mis en évidence la très large gamme de temps de réponse pour les différents éléments du système. Cette forte diversité nous oblige à diviser le cas de test en 2 parties pour permettre la faisabilité des modèles et des bases de données associées. Pour ce modèle, la réponse thermique en fonction de la puissance injectée se révèle totalement linéaire puisque la convection est réduite à un coefficient d'échange. Cependant, si la vitesse de l'air dans les ailettes varie, alors cette simplification n'est plus possible. Pour le modèle du composant, nous avons constaté que la non-linéarité dans la réponse provenait de la dépendance de la conductivité thermique vis-à-vis de la température pour le semi-conducteur utilisé. Toutefois, cette non-linéarité reste limitée dans son amplitude et dans son influence à long terme. Ces constatations vont nous guider dans le choix des méthodes de mesures à utiliser et des méthodes statistiques à sélectionner.

2

Mesures de température

2.1 Introduction

La modélisation numérique nous a permis d'estimer les ordres de grandeurs des différents phénomènes thermiques. Toutefois, ces simulations réclament énormément de temps et de puissance de calcul, notamment pour les modèles à très petite échelle où la différence entre le temps de calcul nécessaire et la durée de fonctionnement simulée est très élevée (exemple : pour le modèle du transistor, 4 jours de simulation pour 1 seconde de signal). De plus, la fiabilité de ces simulations repose uniquement sur la précision des connaissances disponibles sur le système étudié, connaissances qui ne sont pas toujours accessibles (exemple : architecture complète d'un composant électronique). Ces lacunes font que les simulations doivent souvent être complétées par des mesures pour obtenir des bases de données fiables. Ce chapitre propose donc un tour d'horizon des méthodes de mesure de température existantes en présentant leurs principales caractéristiques. Ces caractéristiques seront ensuite comparées afin de dresser un bref comparatif et nous permettre de choisir les méthodes les plus adaptées en fonction des contraintes relevées pour les différentes parties du système. Enfin, les résultats obtenus pour les méthodes de mesure retenues seront présentés.

2.2 État de l'art sur les méthodes de mesure thermique

Cette partie dresse un bref panorama des principales méthodes de mesure thermique, avec leur avantages et leurs inconvénients ainsi que leurs adéquations à notre application.

2.2.1 Rappels sur la propagation des rayonnements électromagnétiques

Une grande partie des méthodes de mesures de température font intervenir des variations de comportement des objets considérés vis-à-vis des rayonnements électromagnétiques. Il est donc utile de rappeler ici quelques notions sur le comportement de ces rayonnements. Dans un milieu transparent tel que l'air, l'onde électromagnétique se propage en ligne droite. À la surface de séparation avec un autre milieu, plusieurs phénomènes peuvent se produire. L'onde électromagnétique peut être renvoyée vers le milieu transparent, on parle alors de réflexion. Si le second milieu est lui aussi transparent, l'onde peut être transmise au travers de ce milieu. Enfin, une partie de la puissance de l'onde peut être transformée en un autre type d'énergie, notamment sous forme d'agitation thermique, et l'on parle alors d'absorption. À chacun de ces phénomènes est associé un coefficient (respectivement réflectance r , transmittance τ et absorptance α) qui représente la proportion de la puissance de l'onde incidente qui va être mise en jeu dans chacun des trois phénomènes (le phénomène de diffusion est volontairement ignoré car il n'intervient

pas dans l'équation-bilan). Il est alors évident que l'on a la relation :

$$r + \tau + \alpha = 1$$

La mesure de température s'effectue le plus souvent sur un milieu qui est considéré comme opaque. La transmittance est donc nulle et le comportement global du milieu vis-à-vis du rayonnement électromagnétique peut être connu en mesurant un seul des paramètres manquants (réflectance ou absorptance). L'absorptance est liée à une autre caractéristique du matériau appelée émissivité. En effet, la plupart des corps ne peuvent pas être considérés comme des corps noirs. Cependant, une approximation de leur comportement peut être obtenue par le modèle dit du corps gris et leurs propriétés émissives sont alors quantifiées par une fraction de celles du corps noir, appelées émissivité. Par la loi de Kirchhoff, dans une enceinte fermée à l'équilibre thermodynamique, les flux radiatifs absorbés et émis vont tendre eux aussi à s'équilibrer. L'émissivité et l'absorptance sont alors égales [Che99].

2.2.2 Mesure de température par infrarouge

Comme nous l'avons vu, la plupart des corps réels peuvent être approximés par des corps gris. Si l'on connait l'émissivité d'un objet, alors en mesurant le flux radiatif émis, il est possible de déterminer la température de cet objet. Ce flux est mesuré à l'aide de photodétecteurs qui transforment le rayonnement électromagnétique en un courant ou une tension électrique. Cependant, la plupart des photodétecteurs ne sont sensibles qu'à une plage de fréquences réduite. Or, aux températures considérées (entre 0 et 1000°C), la majorité du rayonnement électromagnétique du corps noir (et donc des corps gris) s'effectue dans une gamme de fréquences correspondant à l'infrarouge. Pour mesurer la température des objets à partir du flux radiatifs émis, ce sont donc des photodétecteurs sensibles aux infrarouges qui sont utilisés. Les réponses des différents photodétecteurs utilisés usuellement peuvent être trouvées dans la figure 2.1. Lors de la mise en oeuvre de ces mesures, afin d'augmenter l'intensité du signal obtenu mais aussi de réduire le rayonnement infrarouge lié aux réflexions, il est nécessaire d'obtenir la meilleure émissivité possible sur la surface des objets mesurés. L'utilisation de différents type de revêtements (notamment dans le cas de métaux polis) peut alors s'avérer nécessaire. De plus, la mesure infrarouge est également limitée par les longueurs d'ondes utilisées qui restreignent la résolution spatiale minimale accessible.

2.2.3 Mesure de température par thermoréflectivité

La thermoreflectivité désigne la variation de la réflectance induite par la variation de température. Cette relation est souvent mise sous la forme d'une équation linéaire :

$$\frac{\Delta r}{r} = k\Delta T$$

où T représente la température et k le coefficient de thermoréflectivité. Si l'on suppose le coefficient k connu, il suffit donc de mesurer la variation de réflectance pour remonter à la variation de température. L'objet considéré est donc éclairé par une lumière dont la longueur d'onde et l'intensité sont connues. La puissance lumineuse est ensuite mesurée par un photodétecteur pour déterminer la variation Δr (le schéma de principe est illustré à la figure 2.2). Toutefois, le choix de la gamme de longueurs d'onde utilisée n'est ici guidé que par le type de surface considérée et l'on utilise le plus souvent de la lumière visible où les photodétecteurs sont beaucoup plus courants. De plus, les longueurs d'onde étant plus faibles que pour l'infrarouge, la résolution spatiale accessible est plus élevée. Ce principe de mesure fait aujourd'hui l'objet de nombreux travaux, parmi lesquels on pourra notamment citer ceux de Filloy et al. [FTH+03]. Toutefois, dans le cas de l'utilisation d'une caméra pour réaliser la mesure, les

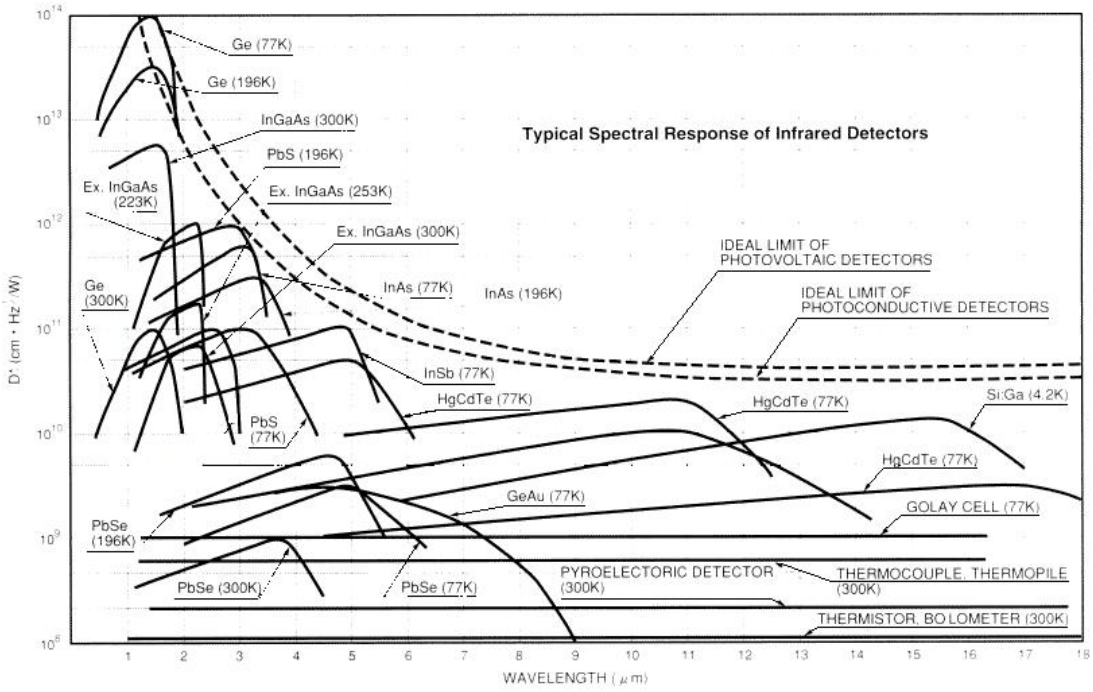


FIGURE 2.1 : Réponses des différents photodétecteurs infrarouges.

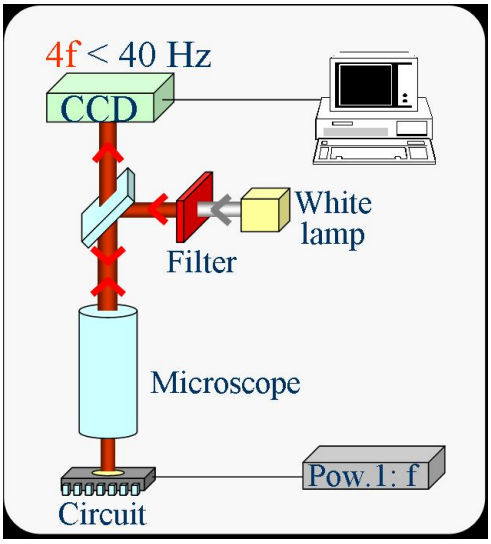


FIGURE 2.2 : Schéma de principe d'une mesure par thermoréflectivité.

fréquences mesurables ne dépassent pas quelques dizaine de Hertz. L'extension de cette méthode pour des signaux thermiques de plus hautes fréquences a été développée grâce notamment à l'utilisation de laser [EDG+05].

2.2.4 Mesure de température par diffusion Raman

La diffusion Raman ou effet Raman a été découvert par le physicien indien Chandrashekhara Venkata Râman. Dans le cas de la mesure de température, la diffusion étudiée est principalement liée à la réflexion. Cette diffusion provient des vibrations du réseau cristallin. En effet, l'effet Raman produit des photons réfléchis qui n'ont pas exactement la même longueur d'onde que les photons incidents. Cette différence est liée à l'interaction entre le photon et l'une des particules en vibration au sein du réseau. Si le photon perd de l'énergie au cours de l'échange, sa longueur d'onde sera plus grande et il correspond alors à une raie dite Stokes dans le spectre lumineux réfléchi.

Si le photon gagne de l'énergie, sa longueur d'onde sera plus courte et il correspond alors à la raie dite anti-Stokes. L'intensité de ces deux raies (ou plutôt le rapport des intensités entre ces deux raies) dépend uniquement des différents modes vibrationnels au sein du réseau cristallin et donc de la température du solide. En observant le spectre réfléchi par une surface, il est donc possible de déterminer sa température. Le principal inconvénient de cette méthode de mesure est la faible intensité des raies Raman. Tout d'abord, cela constitue un problème pour distinguer ces raies à côté de la raie Rayleigh, qui correspond à une diffusion sans échange d'énergie et qui a une intensité beaucoup plus forte. De plus, cela implique des temps de mesure plus longs afin d'obtenir des résultats fiables. Les propriétés de la diffusion Raman et notamment son lien avec la température sont étudiées depuis de nombreuses années et plusieurs articles traitent des applications récentes aux composants électroniques (notamment [ADJC+04]).

2.2.5 Mesure de température par cristaux liquides

Les cristaux liquides sont dans un état intermédiaire entre le réseau cristallin d'un solide et le simple liquide. La forme des molécules est alors très particulière et entraîne un comportement directionnel collectif. Les cristaux liquides cholestériques ont des propriétés optiques qui dépendent fortement de la température. Il peut notamment s'agir d'un changement de couleur. L'utilisation pour la mesure de température est alors relativement simple puisqu'il suffit de relever la couleur d'un matériau recouvert de cristaux liquides pour déterminer sa température. La résolution spatiale de cette méthode est limitée par le type de rayonnement électromagnétique utilisé. Toutefois, comme la surface mesurée doit obligatoirement être recouverte de cristaux liquides, cette méthode est le plus souvent classée dans les méthodes destructives, c'est-à-dire qui altère définitivement les propriétés des objets concernés. De plus, la plage de température accessible se révèle plus limitée que pour les autres méthodes car elle ne dépasse que rarement les 100˚C.

2.2.6 Mesure de température par thermocouples

Les thermocouples sont les capteurs de température les plus employés dans le domaine industriel. Leur principe repose sur l'effet Seebeck. Lorsque deux métaux différents sont reliés par deux jonctions, il se créé alors une tension, appelée tension de Seebeck, dont la valeur varie en fonction de la température. La variation de la tension aux bornes du circuit formé par les deux métaux permet alors de déterminer la différence de température entre les deux jonctions. Ce type de mesure nécessite un contact avec la zone mesurée et ne peut donc pas être appliquée dans le cas d'un composant électronique (sauf dans le cas des thermocouples intégrés aux composants). De plus, la taille des thermocouples reste relativement importante, ce qui peut limiter encore leur utilisation et peut influer sur le temps de réponse du capteur. Les thermocouples sont utilisés depuis de nombreuses années à un niveau industriel pour leur très bonnes performances mais aussi pour leur faible coût et leur « embarquabilité ».

2.2.7 Mesure de température par caractérisation électrique

La plupart des éléments dont on souhaite déterminer la température sont des composants électroniques. Or les propriétés électriques des matériaux sont sensibles aux variations de température. Une méthode pour déterminer la température d'un composant consiste donc à mesurer ses caractéristiques électriques. La caractérisation électrique d'un composant nécessite cependant un scénario de fonctionnement précis, et le fonctionnement normal du composant doit donc être interrompu. De plus, la mesure de température obtenue tient compte de la répartition globale de la température dans l'intégralité de la zone active du composant et non uniquement de la température au niveau jonction. Toutefois, la simplicité d'ajout de cette méthode dans le cadre d'une caractérisation électrique plus vaste fait qu'elle présente un intérêt et qu'elle a notamment fait l'objet d'un brevet [TC02].

Nom	Temps de réponse	Résolution spatiale	Résolution en température	Plage de température	Destructif Destructif
Infrarouge	10μs	15 μ m	0.01˚C	0-1000˚C	Oui/Non
Thermoreflectivité	5 ns	500 nm	0.001˚C	0-1000˚C	Non
Raman	10s	500 nm	0.0001˚C	0-1000˚C	Non
Cristaux liquide	10ms	500 nm	0.1˚C	$\approx$ 0-100˚C	Oui
Electrique	1μs	Zone active du composant	1˚C	0-200˚C	Non
Thermocouple	100ms	Quelques millimètres	0.01˚C	0-1000˚C	Non

TABLE 2.1 : Tableau récapitulant les différentes techniques de mesures thermiques avec leurs principales caractéristiques.

2.2.8 Tableau récapitulatif

Un récapitulatif des caractéristiques des différentes méthodes de mesure est présenté dans le tableau 2.1. Les données proviennent soit des mesures effectuées, soit de l'article de Altet et al. [ACDR06]. Pour la mesure de température sur le système de refroidissement, les contraintes sur les temps de réponse et le positionnement des capteurs sont relativement faibles. C'est donc la praticité des différentes méthodes qui joue un grand rôle. Dans ce cadre, les thermocouples et la mesure IR à l'aide d'une caméra sont les deux méthodes traditionnellement plébiscitées. Afin de s'affranchir des contraintes de la calibration, pour ce projet, le choix s'est porté sur une maquette instrumentée à l'aide de thermocouples. Pour la mesure de température sur le composant, plusieurs méthodes remplissent les conditions requises (microscopie infrarouge, thermoréflectivité ou caractérisation électrique). Toutefois, pour des raisons de praticité et de disponibilité des matériels, c'est la mesure infrarouge qui a été retenue.

2.3 Dispositifs de mesure retenus

Dans le cadre du projet régional où prend place cette thèse, le développement d'une maquette expérimentale destinée à recueillir les données nécessaires a été confié au CORIA (COmplexe de Recherche Interprofessionnel en Aérothermochimie), laboratoire situé à Rouen. Cette maquette s'appuie sur la reproduction du cas de test défini précédemment (composant seul sur un système de refroidissement) et un ensemble de thermocouples.

2.3.1 Réalisation de mesures sur la maquette expérimentale

2.3.1.1 Description de la maquette

Avant de détailler les mesures effectuées, rappelons brièvement le schéma synoptique de la maquette du CORIA. La figure 2.3 suivante illustre schématiquement le principe de commande du composant et le système d'acquisition de données.

Pour simuler le comportement d'un transistor de puissance, des impulsions thermiques sont produites par effet Joule dans une résistance de 50 Ω de type TO220. Ce composant, qui peut fonctionner jusqu'à 150^oC, doit générer des impulsions thermiques dont la durée peut varier de 10 à 500 μs (nominal : 200 μs) et dont la période peut s'étendre de 100 à 5000 μs (nominal : 2000 μs). La puissance instantanée de ces impulsions est fixée à 30 W, ce qui correspond à la dissipation d'un flux thermique de densité égale à $200kWm^{-2}$ à la base du composant. Ce paramètre est conforme aux puissances dissipées dans les transistors de puissance pendant leur fonctionnement.

La génération de signaux de commande et l'acquisition des signaux (tension, courant, températures) sont générées simultanément par une carte multi-fonctions National Instruments de type PCI-6251. Un éditeur graphique commande les trains d'impulsions injectés dans la résistance. La carte gère des signaux à une fréquence de base

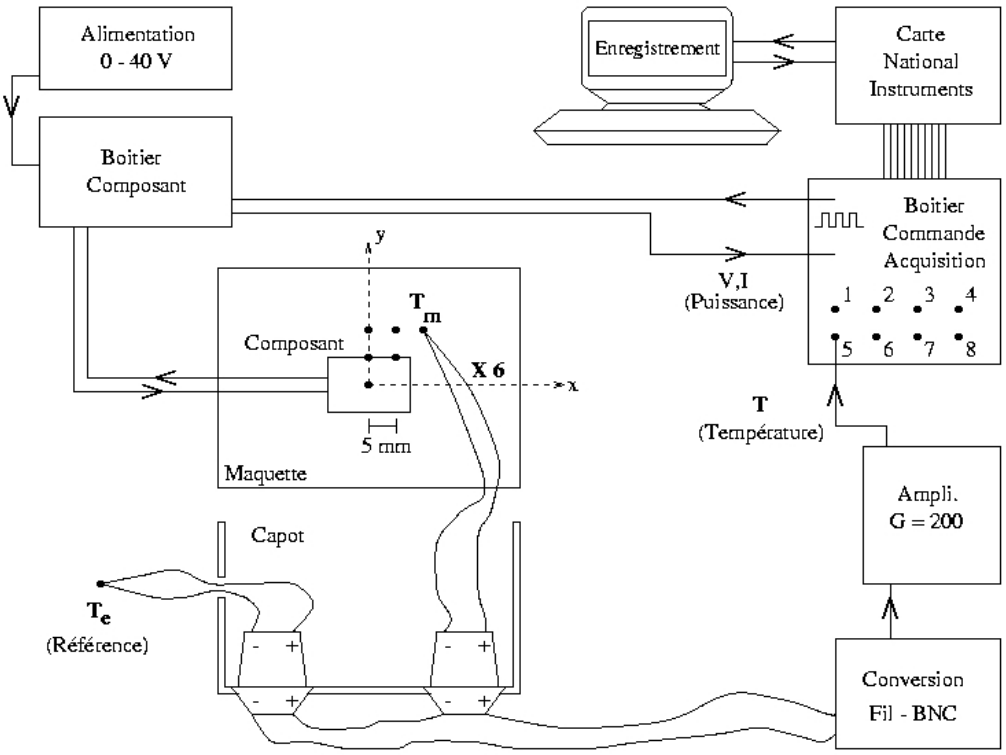


FIGURE 2.3 : Schéma synoptique de la maquette expérimentale du CORIA destinée à relever des mesures de température pour un composant de type résistance équipé d'un système de refroidissement et alimenté par des impulsions de puissance.

égale à 100 kHz, ce qui garantit un temps de 10 μs entre front bas et front haut. Les entrées (acquisition multi-voies) et les sorties (commande du générateur) fonctionnent simultanément. Les signaux des thermocouples (type K) sont amplifiés (Gain x200) et transformés par un polynôme d'interpolation en degrés Celcius. Ces données sont ensuite exportées pour le post-traitement et la visualisation.

Chaque mesure correspond à l'enregistrement simultané de huit grandeurs : six températures égales à la différence entre la température au point considéré du support du composant et la température ambiante $T(x,y,t) = T_m(x,y,t) - T_e$, la tension V et l'intensité I du courant traversant le composant. Ces six points de mesure, notés M, R, J, V, G, et B, sont repérés dans un système orthonormé (x, y) centré sur le composant. Ox est la dimension longitudinale des ailettes et Oy la dimension transversale. L'unité de longueur étant le millimètre, les coordonnées de ces points sont les suivantes : M (0,0), R (0,5), J (5,5), V (0,10), G(5,10) et B (10, 10). L'intérêt de ces points de mesure est d'analyser l'évolution temporelle de la température interne de la résistance (point M) mais aussi l'évolution spatio-temporelle de cette température en prélevant des mesures en des points situés à proximité de M. De plus, ils permettent d'évaluer l'intérêt du positionnement d'un capteur à proximité d'un composant afin d'aider à la prédiction de sa température.

La maquette est placée dans une veine rectangulaire, en plexiglas. Elle est reliée à un ventilateur par une connexion en PVC. Deux passages ont été pratiqués dans la paroi de cette veine pour y faire coulisser des sondes anémométriques. On peut ainsi, par déplacement transversal de ces sondes, établir les profils de vitesse et de température de l'air utilisé dans le système de refroidissement en amont et en aval de la maquette.

2.3.1.2 Résultats expérimentaux

Nous avons étudié dans un premier temps le comportement de la maquette dans le cas statique. La température sous le composant (au point M) a donc été relevée pour une puissance et une vitesse d'air dans les ailettes constantes (figure 2.4 et 2.5). Ces courbes illustrent bien la réponse non-linéaire de la température vis-à-vis de la vitesse de l'air dans les ailettes. En effet, chacune de ces vitesses va créer des conditions d'écoulement de l'air différentes dans les ailettes. Si le coefficient d'échange à la surface des ailettes augmente effectivement en fonction de la vitesse de l'air, la relation n'est cependant pas linéaire. En revanche, pour une vitesse d'air fixée, le coefficient d'échange reste constant et pour un scénario statique, le système de refroidissement peut alors être assimilé à une résistance thermique. L'élévation de température est ainsi directement proportionnelle à la puissance dissipée dans le composant.

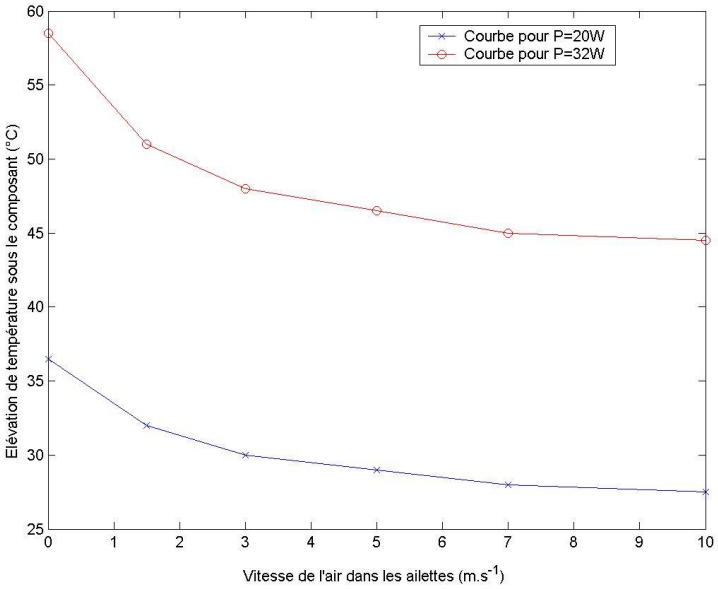


FIGURE 2.4 : Évolution de la température sous le composant en fonction de la vitesse de l'air dans les ailettes pour une puissance injectée fixée.

Différents scénarios de rafales et de vitesses de refroidissement ont ensuite été programmés, en faisant varier un seul des paramètres ou les deux de manière asynchrone (figure 2.7) et la température sous le composant a été relevée (figure 2.6 et figure 2.8). Les thermocouples ne permettent cependant pas de suivre la réponse thermique à chaque impulsion de puissance (quelques centaines de microsecondes). En effet, bien que les thermocouples soient placés jusque sous le composant (ce qui n'est pas possible pour une application industrielle), la température qu'ils mesurent est principalement celle du système de refroidissement et non pas celle du composant. Ce phénomène est lié au fait que les thermocouples sont placés dans des encoches percées dans le support en aluminium, qui possède une très bonne conductivité thermique. Ceci est d'autant plus vrai que le thermocouple est placé loin du composant, la chaleur étant un phénomène diffusif. Pour la suite de l'étude, seule la température du point M est donc utilisée. Toutefois, même au point M, le temps de réponse du signal mesuré est de l'ordre d'une seconde. Pour les dynamiques les plus lentes, la vitesse de l'air dans les ailettes semble avoir une légère influence sur l'établissement de la température d'équilibre (figure 2.9). Toutefois, afin de conserver des bases de données de tailles raisonnables, les échelons utilisés pour la puissance injectée ou la vitesse de l'air dans les ailettes ne dépassent pas 5 minutes.

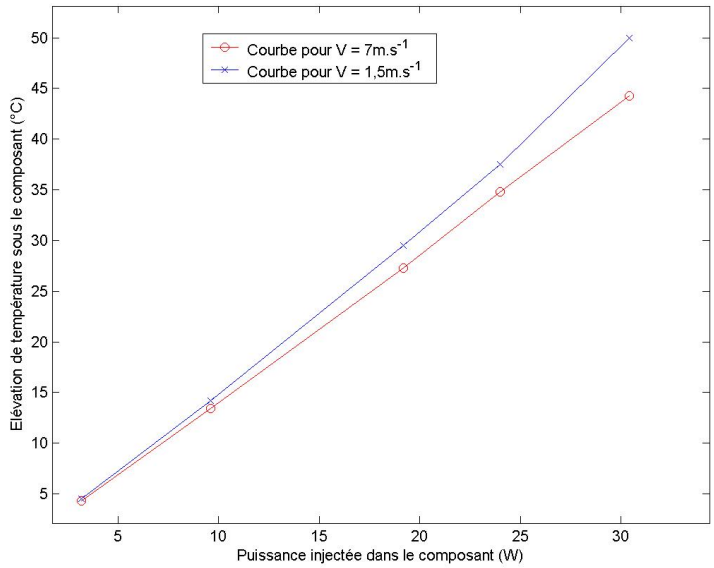


FIGURE 2.5 : Évolution de la température sous le composant en fonction de la puissance injectée pour une vitesse d'air dans les ailettes fixée.

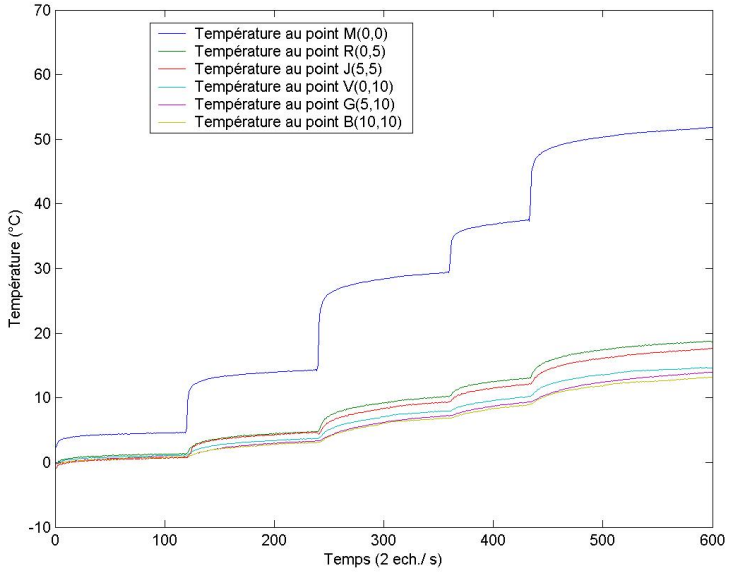


FIGURE 2.6 : Température relevée pour les différents thermocouples pour des échelons de puissances croissantes et une vitesse d'air dans les ailettes de $1,5 m.s^{-1}$.

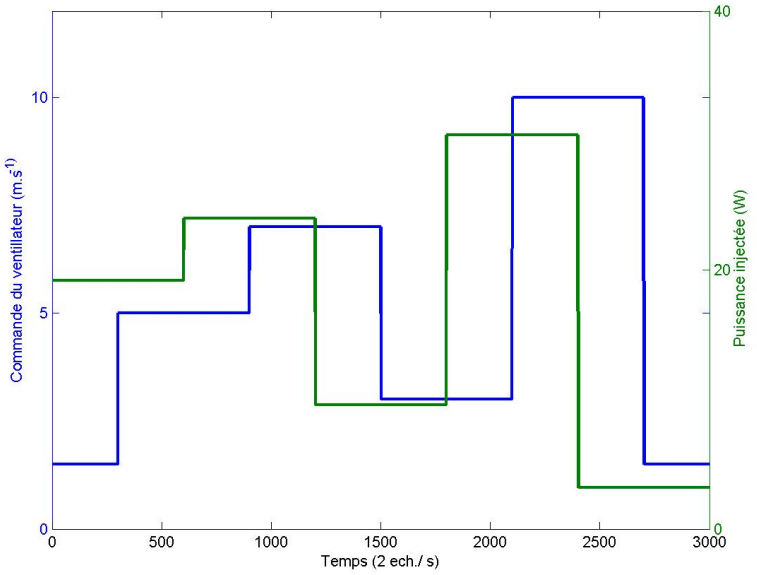


FIGURE 2.7 : Scénario de puissance injectée et de vitesse d'air dans les ailettes pour la maquette expérimentale.

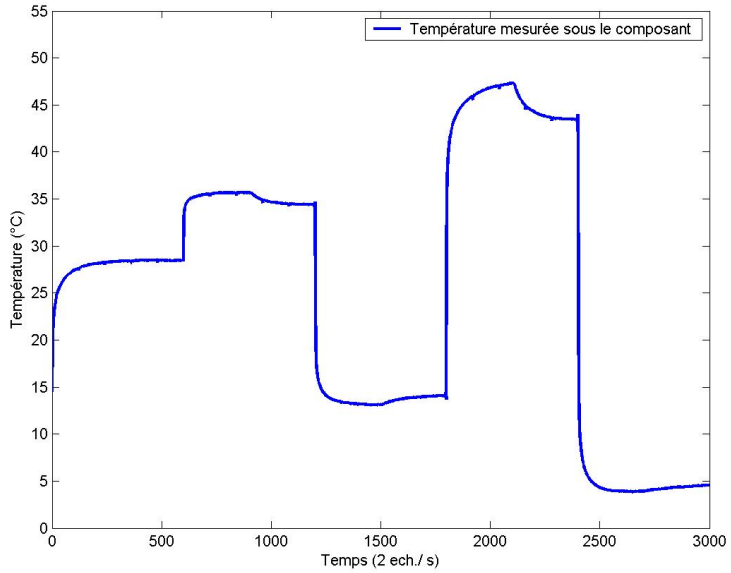


FIGURE 2.8 : Température relevée sous le composant pour le scénario considéré à la figure 2.7.

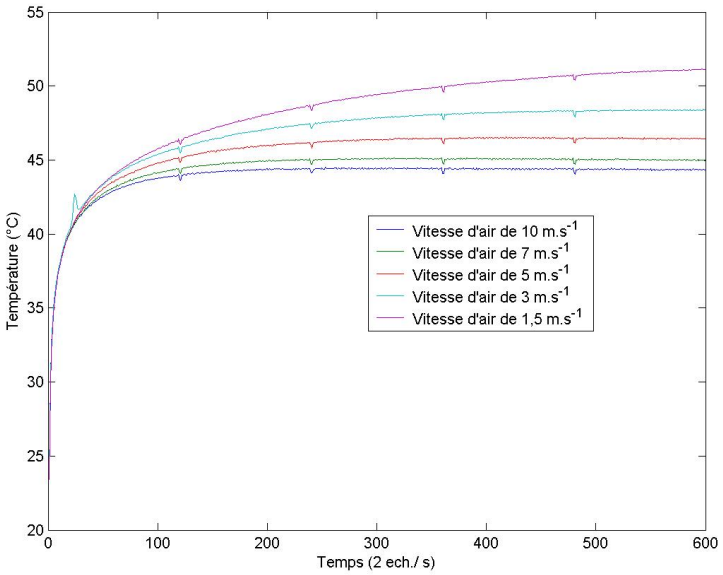


FIGURE 2.9 : Courbes de température pour un échelon de puissance de 32W avec différentes vitesses d'air dans les ailettes.

2.3.2 Mesures infrarouges

Les données de la maquette permettent donc de suivre l'évolution de la température moyenne du composant au cours du temps. Toutefois, pour les dynamiques les plus rapides, il nous a fallu compléter nos données à partir d'un autre type de mesure. Pour cette tâche, en fonction de leurs capacités mais aussi de la disponibilité des équipements de mesure, le choix des méthodes infrarouges est apparu comme le plus pertinent. Deux dispositifs sont utilisés pour localiser puis suivre au cours du temps la température la plus élevée à la surface du composant.

2.3.2.1 Caméra infrarouge

Le fonctionnement de la caméra infrarouge repose sur l'utilisation d'une matrice non refroidie bidimensionnelle de capteurs micrométriques appelés microbolomètres. Au sein de chaque microbolomètre, les radiations infrarouges émises depuis le système à thermographier sont captées par un matériau absorbant. La variation de température qui en résulte va induire une variation de résistance électrique qui peut être mise sous la forme d'un signal électrique. Le signal de température analogique amplifié est ensuite converti en un signal numérique, qui est représenté par une image thermique en fausses couleurs (figure 2.10). Cette méthode permet de mesurer efficacement le fonctionnement d'un système de refroidissement ou même la température moyenne d'un composant. Toutefois, les faibles résolutions spatiale ($100\mu m$ de résolution maximale avec un objectif grossissant) et temporelle (30Hz) ne permettent pas de suivre les variations de température dans un transistor au cours d'une impulsion.

Dans le cadre des mesures de températures sur des circuits électroniques, l'utilisation de l'imagerie infrarouge (IR) permet un diagnostic pratique et rapide. Toutefois, afin de fournir une mesure plus précise, il est nécessaire de prendre en compte l'émissivité des différents matériaux. Déterminer a priori cette émissivité s'avère malheureusement impossible, et l'on a recourt à des mesures de calibration (où la température est fixée par un support chauffant) afin de l'estimer. Dans ce cadre, l'approche classique consiste à calculer l'émissivité moyenne sur des zones homogènes et à reporter manuellement ce résultat sur les images de mesures. Cependant, le nombre de mesures peut être très important, et ce type de tâche devient donc très long et fastidieux. L'automatisation du recalage de l'émissivité permet donc un gain de temps non négligeable dans le traitement de séries d'images IR, mais elle nécessite un repositionnement précis entre les images de mesures et celles de calibration (du fait du déplacement

de la caméra entre les 2 séries d'images). Les techniques récentes de traitement d'image et des statistiques sont utilisées pour estimer les différents paramètres de ce repositionnement.

La transformation géométrique entre l'image de calibration et l'image de mesure est supposée affine. Elle est donc composée des 3 similitudes : homothétie, rotation et translation. Pour repositionner les 2 images, il faut donc déterminer 4 paramètres : angle de rotation, facteur d'échelle, déplacements vertical et horizontal. On pose les contraintes suivantes pour la précision attendue : 1 degré pour l'angle de rotation, 0.01 pour le facteur d'échelle et 1 pixel pour les 2 déplacements.

La méthode choisie consiste à déterminer les 4 paramètres de manière successive. En effet, en utilisant une extraction des contours de l'image par la méthode de Canny et une détection des coins par la méthodes de Harris, il est possible de calculer l'angle de rotation et le facteur d'échelle qui existent entre l'image de calibration et l'image de mesure. Les paramètres de la translation peuvent alors être retrouvés très facilement par corrélation. Une fois les 2 images superposées, le recalage d'émissivité devient aisé. A partir de l'image de calibration, la température étant connue, il est possible de retrouver l'émissivité sur chaque pixel. Cette émissivité est ensuite reportée sur l'image de mesure afin de corriger les températures relevées. Cette méthode, que nous avons développée, a fait l'objet d'un dépôt de brevet [MP09].

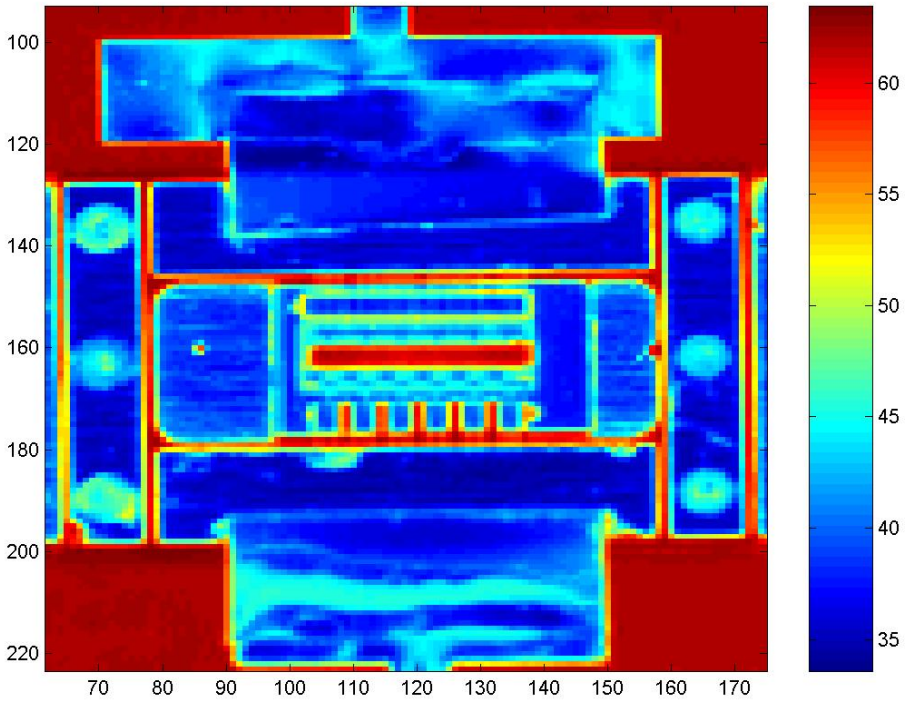


FIGURE 2.10 : Exemple de mesure effectuée avec une caméra infrarouge.

2.3.2.2 Microscope infrarouge

Ces mesures sont effectuées par l'intermédiaire d'un microscope infrarouge de type Barnes RM-2A. Le rayonnement émis par le corps à mesurer est focalisé par l'intermédiaire d'un groupe de lentilles vers un capteur sensible au rayonnement infrarouge (illustration dans le schéma 2.11). Ce détecteur infrarouge produit une faible tension proportionnelle à la luminance du corps rayonnant. Le signal ainsi recueilli permet de remonter à la température surfacique ou à l'émissivité du composant. La tension aux bornes du détecteur étant faible, elle est amplifiée par

un amplificateur à bande étroite de façon à obtenir un gain important. De plus, afin de diminuer au maximum le niveau de bruit thermique, le capteur IR est refroidi par de l'azote liquide. La mesure de température est ponctuelle avec un spot d'environ 30μ m.

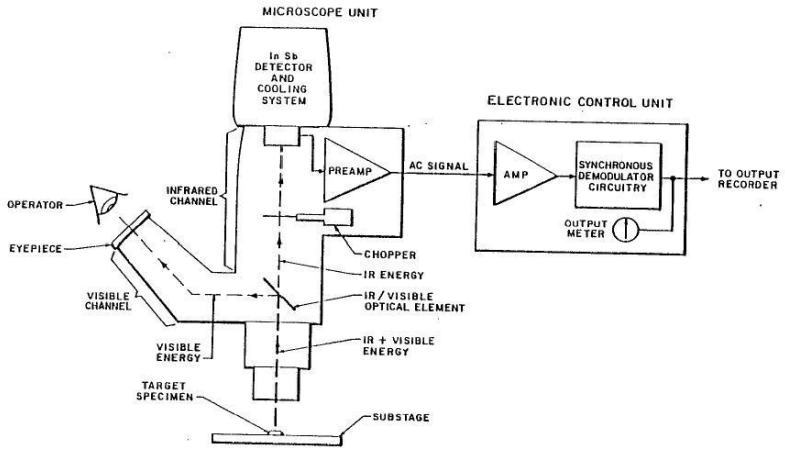


FIGURE 2.11 : Schéma de principe du microscope infrarouge RM-2A.

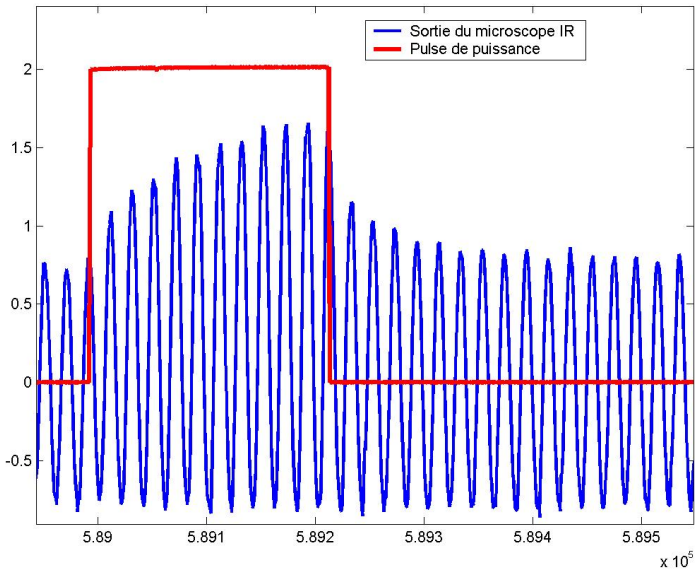


FIGURE 2.12 : Mesures dynamiques de la température. En rouge, l'impulsion de puissance et en bleu, la sinusoïde modulée par la variation de température.

Le signal électrique recueilli est constitué d'une sinusoïde dont l'amplitude dépend de l'énergie captée par le détecteur comme l'illustre la figure 2.12. À partir de ce signal électrique, plusieurs solutions sont envisageables pour retrouver le véritable signal thermique, qui est constitué par l'enveloppe de cette sinusoïde. Jusqu'à présent, la méthode employée était d'utiliser une superposition temporelle sur plusieurs trames de impulsions. En effet, si le composant est arrivé dans un régime dit pulsé-stabilisé (la valeur moyenne de la température pour chaque pulse ne change plus) alors la superposition des mesures obtenues pour plusieurs périodes d'impulsions permet de reconstruire l'évolution complète de la température au cours d'un pulse (puisque chacune des périodes est en fait la répétition du même « signal de température »).

Bien sûr, les méthodes classiques de démodulation sont également utilisables (notamment l'utilisation d'un filtre passe-bas). Pour une utilisation optimale de ces méthodes, il est toutefois nécessaire de déterminer tout d'abord la fréquence de la porteuse utilisée par le microscope IR. Or, il peut être démontré que l'estimateur optimal au sens des moindres carrés de la fréquence d'une sinusoïde représentée par un nombre fini d'échantillons est obtenu par le maximum de la transformée de Fourier en temps discret du signal [Kay93]. Une analyse fine de la fréquence de la porteuse du microscope IR a ainsi permis de constater que cette fréquence se modifiait au cours du temps. L'utilisation de méthodes d'asservissement de phase est donc nécessaire pour obtenir une pseudo-sinusoïde parfaitement synchrone avec la porteuse du microscope IR. Un travail important reste cependant à effectuer pour extraire le maximum d'information dans le cas où la période des impulsions de commande se rapproche de le fréquence de la porteuse du signal.

2.4 Conclusion

Comme nous l'avons vu lors des simulations, compte tenu de la grande différence entre le temps de réponse du composant (quelques dizaine de μs) et du système de refroidissement (plusieurs minutes), il est nécessaire de scinder en deux notre système afin d'être capable d'obtenir des données mais aussi de conserver des bases de données de tailles raisonnables. Pour le système de refroidissement, la maquette du CORIA permet d'obtenir les données nécessaires à l'établissement d'un modèle thermique. De plus, le très fort taux d'échantillonnage utilisé (100Hz) par rapport aux dynamiques accessibles, de l'ordre de la seconde, permet d'effectuer un filtrage des mesures qui élimine presque totalement le bruit sans perte d'information sur les dynamiques présentes dans le signal. Pour ce cas, les données utilisées seront issues des scénarios où la vitesse de l'air dans les ailettes et la puissance injectée varient.

Pour le composant, à l'heure actuelle, la méthode utilisée pour les mesures au microscope infrarouge ne permet de relever l'évolution de la température que pour un type de pulse précis (période et rapport cyclique fixés). Or ce cas n'est pas adapté à l'apprentissage statistique car les méthodes que nous envisageons de tester nécessitent théoriquement la condition d'excitation persistante [Lju02] des impulsions de commande. Cette condition stipule que le signal d'entrée d'un système dynamique doit couvrir une large plage de fréquences afin de garantir une bonne identification de son modèle statistique. Autrement dit, le signal d'entrée doit avoir des caractéristiques proches du bruit blanc ou du moins avoir la forme d'une séquence binaire pseudo-aléatoire (SBPA) pour remplir cette condition. Cependant, la génération de bases de données plus complexes nécessite des systèmes de génération de la commande des composants plus sophistiqués, qui ne sont pas encore disponibles. Les données relevées permettent néanmoins de déterminer le type de bruit présent sur les mesures au microscope IR comme le montre la figure 2.13. Ce bruit est de type gaussien et ses paramètres peuvent être calculés. La solution utilisée est donc de reporter ce bruit sur les données issues des simulations obtenues pour le modèle complet du composant 1.6.2 afin d'obtenir une base de données réaliste. Cet ajout de bruit permet d'obtenir un signal exploitable pour les méthodes d'apprentissage sans pour autant être trop optimiste sur les capacités des méthodes de mesure qui devront être utilisées pour recueillir de telles données.

Au final, on dispose donc de deux jeux de données différents qui représentent la quasi-totalité des dynamiques rencontrées pour le refroidissement d'un composant électronique dans un radar. Le premier de ces jeux caractérise la réponse thermique d'un transistor de puissance. Il s'agit de données issues de simulation. Le signal d'entrée est un signal binaire pseudo-aléatoire avec cependant une contrainte sur les durées minimales et maximales des créneaux de puissance (de 100 μs à 1000 μs), ce qui correspond aux pulsations rencontrées lors d'un fonctionnement normal du composant. Les signaux d'entrée et de sortie sont échantillonnés à 10000Hz sur une durée d'une seconde. Le bruit identifié sur les mesures au microscope infra-rouge (rapport signal/bruit de 10) est ensuite reporté sur les mesures de sortie. Le second jeux de données est constitué par les mesures effectuées sur la maquette du CORIA. Cinq scénarios de variations de la puissance dissipée et de la commande du ventilateur ont été utilisés. Ces scénarios sont d'une durée de 40 minutes et sont élaborés de sorte que les entrées (puissance dissipée dans le

composant et vitesse de l'air dans les ailettes) soient des créneaux d'amplitude et de durée différentes. La température sous le composant a été relevée à une fréquence de 100Hz puis filtrée par un filtre passe-bas pour ne conserver finalement qu'un échantillon sur 100. Le récapitulatif des bases de donnée disponible est donné dans la table 2.2.

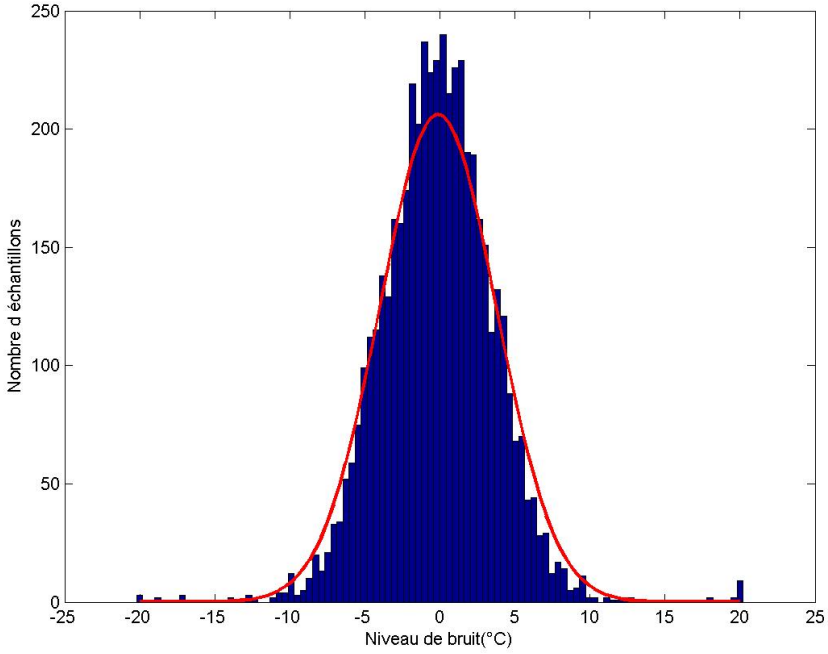


FIGURE 2.13 : Histogramme du bruit affectant les mesures effectuées à l'aide d'un microscope IR. Cet histogramme peut être approché par une loi gaussienne.

Nom	Durée	Fréquence d'échantillonnage	Nombre de points	Nombre de séries
Transistor	1 s	10000 Hz	10000	1
Maquette	40 min	1 Hz	2400	5

TABLE 2.2 : Tableau récapitulatif des bases de données thermiques utilisées pour la modélisation statistique.

3

Etude et développement de modèles statistiques d'un système thermique radar

3.1 Introduction

L'apprentissage statistique est le plus souvent utilisé lorsque les connaissances a priori disponibles sur un système sont insuffisantes ou lorsque les paramètres du modèle physique issu de la mise en équation du système (comme nous l'avons décrit dans le chapitre 1) ne peuvent être identifiés ou simplement, comme dans le cadre de notre application, lorsque le modèle physique ne se prête pas à la finalité visée, qui est ici une utilisation embarquée. Les seules connaissances restantes sont alors les données entrée-sortie qui ont pu être recueillies par la mesure sur le système. Le but de l'apprentissage automatique est alors de trouver le meilleur modèle mathématique capable de reproduire le comportement du vrai système sur la base de ces données.

Ces possibilités font qu'il existe trois approches de modélisation possible :
- la modélisation « boite blanche » basée sur les principes physiques mis en oeuvre au sein du système à modéliser (illustrée dans le chapitre 1),
- la modélisation « boite noire » consistant à partir de mesures expérimentales à trouver une relation mathématique reliant les entrées du système à ses sorties observées,
- la modélisation « boite grise » [LL95, WG04] qui essaie de tirer parti des avantages des deux premières approches.

Dans le cadre de ce chapitre, nous mettrons l'accent sur la deuxième approche, c'est-à-dire la modélisation statistique d'un système dynamique avec application sur un système thermique. Après une présentation du cadre théorique général de l'apprentissage statistique, nous porterons notre attention sur la modélisation entrée-sortie des systèmes dynamiques puis nous mettrons en évidence les spécificités de cette modélisation dans le contexte général de l'apprentissage. Ces spécificités sont par exemple relatives au choix des dynamiques du système, à la notion de stabilité des modèles identifiés. L'application des approches de modélisation existantes, et ayant fait leurs preuves dans la littérature, au problème de modélisation de la température interne des composants est finalement présentée.

Au cours de ce chapitre, les algorithmes de types réseaux de neurones récurrents, les algorithmes basés sur les méthodes à noyaux et les modèles bayésiens dynamiques seront étudiés. L'intérêt de considérer les réseaux de neurones dans le contexte de notre application est la facilité d'implémentation d'un tel modèle sur des calculateurs embarqués. Le choix d'appliquer les méthodes à noyaux à la prédiction de la température interne des composants se justifie par le fait que ces méthodes ont fait leur preuve dans de nombreuses applications, notamment en classification. Les réseaux bayésiens présentent l'avantage de fournir en plus de la prédiction, un intervalle de confiance sur l'estimation. Cette propriété est intéressante dans une perspective industrielle car elle permet d'associer une signification statistique aux actions effectuées (diminution ou augmentation de la puissance injectée, détection de

la proximité du point de rupture des composants, ...) à partir des prédictions. Pour chaque type de modélisation, nous nous efforcerons de faire ressortir les forces et faiblesses de ces différentes approches par rapport à l'application concernée. Une comparaison des résultats empiriques sur des jeux de données issues des mesures sur la maquette expérimentale et sur des données issues de la simulation est présentée. À des fins de comparaison, les performances de référence fournies par les modèles linéaires (ou pseudo-linéaires) sont également présentées. Le chapitre se termine par une analyse récapitulative de la modélisation statistique.

3.2 Cadre théorique général

On considère une base de données définie par un ensemble de couples entrée-sortie $\{(\mathbf{x}_i, y_i)\}_{i=1,\cdots,N}$ où le vecteur d'entrée est $\mathbf{x}_i \in \mathscr{X}$ et où $y_i \in \mathscr{Y}$ représente la sortie. Le but de l'apprentissage automatique est alors de trouver la meilleure représentation fonctionnelle :

$$\begin{array}{rcl} f : \mathscr{X} & \longrightarrow & \mathscr{Y} \\ \mathbf{x} & \longmapsto & f(\mathbf{x}) \end{array} \tag{3.1}$$

permettant d'expliquer au mieux ces données.

Bien que la préoccupation principale de cette thèse soit la modélisation d'un système dynamique (en l'occurrence la modélisation des variations de la température interne de composants électroniques en fonction des puissances injectées), nous allons dans un premier temps expliquer le cadre général de la problématique d'apprentissage statistique pour nous focaliser par la suite sur les particularités de la modélisation dynamique.

Les différents problèmes d'apprentissage sont généralement regroupés au sein de 2 sous-groupes en fonction des particularités du vecteur de sortie y_i :

- La classification : la sortie y_i est représentée par un ensemble discret fini de valeurs. Le problème consiste alors à attribuer au vecteur $\mathbf{x}_i$ la bonne classe parmi celles disponibles. Le cas particulier de la classification binaire correspond à $\mathscr{Y} = \{0,1\}$ ou $\mathscr{Y} = \{-1,1\}$.
- La régression : la sortie y_i est représentée par une ou plusieurs valeurs réelles (i.e. $\mathscr{X} = \mathbb{R}^p$). Ce dernier cas de figure est celui qui est le plus proche de la modélisation de système que nous décrirons ultérieurement.

Le cas de l'apprentissage non supervisé (absence de la sortie y_i) ou du classement (« ranking ») ne sera pas abordé dans ce document.

3.2.1 Formalisation mathématique du problème d'apprentissage statistique

Pour établir un cadre mathématique à l'apprentissage automatique, on considère que les données d'apprentissage $(\mathbf{x}_i, y_i) \in \mathscr{X} \times \mathscr{Y}$ sont les réalisations d'une variable aléatoire jointe (X,Y) suivant une loi de probabilité $P(X,Y)$. Cette loi associée à une densité de probabilité $p(\mathbf{x},y)$ sur le couple $(\mathbf{x},y)$ est le plus souvent inconnue. On suppose également que la fonction f recherchée appartient à un espace d'hypothèses $\mathscr{H}$. Pour quantifier la qualité des prédictions fournies par f, on définit une fonction de coût mesurant l'écart entre la vraie sortie y et sa prédiction $f(\mathbf{x})$. Cette fonction de coût est définie de façon générique comme :

$$\begin{array}{rcl} \ell : \mathscr{Y} \times \mathscr{Y} & \longrightarrow & \mathbb{R}^+ \cup \{0\} \\ (\mathbf{z}, y) & \longmapsto & \ell(\mathbf{z}, y). \end{array} \tag{3.2}$$

Le meilleur modèle sera alors celui qui minimisera l'espérance de ce coût sur l'ensemble des exemples possibles. Cette mesure, nommée erreur de généralisation ou risque et noté $R(f)$, est donnée par la formule :

$$R(f) = \mathbb{E}(\ell(f(\mathbf{x}), y)) = \int_{\mathscr{X} \times \mathscr{Y}} \ell(f(\mathbf{x}), y) p(\mathbf{x}, y) d\mathbf{x} dy \tag{3.3}$$

où $\mathbb{E}$ désigne l'espérance mathématique. La densité de probabilité $p(\mathbf{x}, y)$ étant inconnue et de plus ne disposant que d'un ensemble fini d'observations, on considère les données équiprobables et on définit le risque empirique par la formule

$$R_e(f) = \frac{1}{N}\sum_{i=1}^{N} \ell(f(\mathbf{x}_i), y_i). \tag{3.4}$$

Le but d'un algorithme d'apprentissage sera alors de trouver la fonction f^* qui minimise le risque empirique :

$$f^* = \underset{f \in \mathscr{H}}{\operatorname{argmin}} R_e(f). \tag{3.5}$$

Cette formulation du problème étant faite, des précisions sur la famille d'hypothèses ainsi que la fonction de coût sont apportées dans les sous-sections ci-dessous.

3.2.1.1 Famille de fonctions

Dans les problèmes d'apprentissage automatique, les modèles linéaires (du type $f(\mathbf{x}) = \mathbf{w}^\top \mathbf{x}$ avec $\mathbf{w}$ le vecteur de paramètres du modèle) sont souvent privilégiés. L'intérêt réside dans leur simplicité qui conduit souvent à des algorithmes d'optimisation efficaces et simples. Toutefois, la plupart des problèmes réels présentent à différents degrés des non-linéarités. Une manière astucieuse de résoudre un problème de modélisation non linéaire consiste à chercher la solution f sous la forme d'une combinaison linéaire au sein d'une famille génératrice $\mathscr{F} = \{f_j \in \mathscr{H}_j\}$ de fonctions non-linéaires. La fonction f est alors de la forme :

$$f(\mathbf{x}) = \sum_j w_j f_j(\mathbf{x}). \tag{3.6}$$

Cette formulation a l'avantage de permettre d'obtenir une fonction linéaire par rapport aux paramètres w_j. Dans certains cas (comme par exemple pour les SVM [Vap95] ou les splines d'interpolation [Wah90]), si les fonctions génératrices sont fixées, il est possible de se ramener à des algorithmes d'optimisation établis dans le cadre linéaire. Cette formulation très générale, dite *à fonctions de base* ([Lju02]), permet d'englober la quasi-totalité des fonctions utilisées dans le cadre de l'apprentissage statistique (réseaux de neurones, SVM, splines, ...).

Dans la suite de ce chapitre 3, nous préciserons dans le cadre des systèmes dynamiques qui nous intéressent, les familles de modèles particulières à considérer et qui permettent de prendre en compte la temporalité des données.

3.2.1.2 Fonction de coût

Le problème d'apprentissage nécessite le calcul un coût mesurant l'écart entre l'estimation du modèle et la sortie réelle correspondante. Nous listons quelques exemples de fonctions de coût pour les problèmes de classification et de régression.

Fonction de coût pour la classification

Les fonctions de coût usuelles dans le cas de la classification binaire sont :

– le coût $0-1$ qui consiste en un simple comptage du nombre d'erreurs

$$\ell(f(\mathbf{x}), y) = \mathbf{1}_{f(\mathbf{x}) \neq y}$$

où la fonction indicatrice $\mathbf{1}_u$ vaut 1 si le prédicat u est vrai et 0 autrement. Cette fonction a le désavantage d'engendrer un problème d'optimisation combinatoire difficile à résoudre. Une solution consiste alors à utiliser des fonctions de coût bornantes et plus faciles à optimiser,

– le coût charnière ou sa version quadratique très prisée dans les algorithmes de type SVM

$$\text{Coût charnière :} \quad \ell(f(\mathbf{x}),y) = \max(0, 1 - yf(\mathbf{x}))$$
$$\text{Coût charnière quadratique :} \quad \ell(f(\mathbf{x}),y) = \max(0, 1 - yf(\mathbf{x}))^2$$

– le coût logistique défini par $\ell(f(\mathbf{x}),y) = \log\left(1 + \exp^{-yf(\mathbf{x})}\right)$.

Une illustration de ces fonctions est portée sur la figure 3.1.

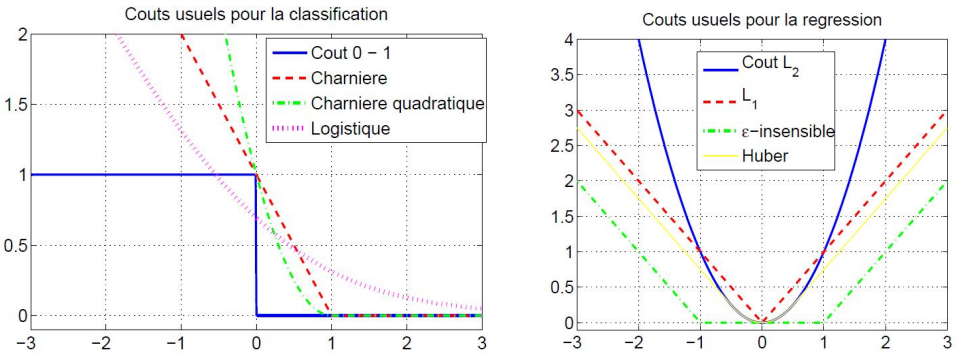


FIGURE 3.1 : Coûts usuels pour les problèmes de classification binaire et pour la régression.

Fonctions de coût pour la régression

Ce sont principalement :

– les coûts ℓ_p normes

$$\ell(f(\mathbf{x}),y) = |y - f(\mathbf{x})|^p \quad \text{avec} \quad p \geq 1.$$

Dans la gamme de ces coûts, on distingue principalement le coût des moindres carrés qui correspond à $p = 2$ (et qui fait l'hypothèse des sorties entachées par un bruit blanc gaussien) et le coût des moindres valeurs absolues obtenu avec $p = 1$ (et qui correspond à l'hypothèse d'un bruit de type Laplace).

– le coût ε-insensible

$$\ell(f(\mathbf{x}),y) = \max(0, \varepsilon - |y - f(\mathbf{x})|)$$

sur lequel repose la méthode des machines à vecteur support pour la régression [SS01].

– le coût de Huber défini par

$$\ell(f(\mathbf{x}),y) = \begin{cases} 2\varepsilon|y - f(\mathbf{x})| - \varepsilon^2 & \text{si} \quad |y - f(\mathbf{x})| \geq \varepsilon \\ (y - f(\mathbf{x}))^2 & \text{autrement.} \end{cases}$$

On fait appel généralement à ce coût pour rendre l'algorithme d'apprentissage robuste aux éventuelles données aberrantes. La visualisation graphique de ces coûts est également portée sur la figure 3.1.

Pour finir avec ce bref panorama, précisons qu'on peut classer ces coûts en catégories en fonction de leur convexité et de leur singularité (voir tableau 3.1). Ces propriétés particulières conditionnent les algorithmes d'optimisation efficaces qui sont appliqués à ces problèmes. La convexité garantit (sous certaines conditions) l'existence d'un minimum unique alors que la singularité va conditionner la robustesse ou la parcimonie de la solution. La différentiabilité des fonctions de coût facilite l'utilisation d'algorithmes de type descente de gradient [Bon06]. Remarquons que même si le coût n'est pas différentiable, dans le cas convexe, on peut toujours mettre en oeuvre des algorithmes d'optimisation basés sur l'utilisation du sous-gradient [HUL93].

	Différentiable	Singulier
Convexe	Coût ℓ_p avec $p > 1$ Coût de Huber Coût logistique Coût charnière quadratique	Coût ε-insensible Coût Charnière Coût ℓ_1
Non-convexe	Coût sigmoide	0-1

TABLE 3.1 : Classification des critères d'apprentissage selon leurs particularités (convexité et singularité).

3.2.2 Contrôle de la complexité du modèle

3.2.2.1 Le sur-apprentissage

Une fois la famille de fonctions et la fonction de coût choisies, la minimisation du risque empirique se résume à un problème d'optimisation. Or, si la famille de fonctions choisie est suffisamment riche (par exemple si le nombre de paramètres libres impliqués dans l'apprentissage dépasse le nombre d'exemples) alors il sera possible de trouver une fonction f^* telle que $R_e(f^*) = 0$. Ce fait peut devenir problématique dans le cas des données bruitées. En effet, le modèle appris par minimisation du risque empirique se sera adapté aux particularités du bruit ayant corrompu les données d'apprentissage, affectant de ce fait les capacités du modèle à estimer correctement les sorties correspondant à de nouvelles données. On parle alors de mauvaise capacité de généralisation. Estimer le risque (3.3) à partir uniquement de l'ensemble d'apprentissage n'est donc pas suffisant et des alternatives doivent être trouvées. Ainsi, la solution consiste à borner la capacité de la famille de fonctions ou à adjoindre une contrainte supplémentaire sur le problème lors de l'apprentissage.

3.2.2.2 Le dilemme biais-variance

Le danger de générer des modèles de capacités excessives et donc la nécessité de contrôler leur complexité est illustré par le dilemme dit de "biais-variance" [GBD92]. Pour illustrer cette problématique, on considère un problème de régression sur des données bruités. Les données sont générées par un modèle réel f_{vrai} et un bruit blanc gaussien ε d'espérance nulle et de variance σ^2 entache les mesures. Les données recueillies $(\mathbf{x}, y)$ sont donc de la forme :

$$y = f_{vrai}(\mathbf{x}) + \varepsilon \tag{3.7}$$

Pour notre exemple, compte tenu des propriétés du bruit ce problème est résolu en utilisant un coût quadratique $\ell(f(\mathbf{x}), y) = (y - f(\mathbf{x}))^2$. On suppose également que l'algorithme d'apprentissage est capable d'obtenir la solution optimale f^* dans la famille de fonctions considérée et quelque soit cette famille de fonctions choisie (cette hypothèse n'est pas toujours vérifiée en pratique du fait d'un nombre limité d'itérations de l'algorithme d'apprentissage choisi ou d'approximations numériques, et l'on obtient une solution approchée $\tilde{f}^*$ [BB08]). Une vue simplifiée de ces différentes erreurs est présentée à la figure 3.2.

On s'intéresse à l'erreur $Err(\mathbf{x})$ commise en un point $\mathbf{x}$ choisi de l'espace des exemples. On note $\mathbb{E}$ l'espérance et $\mathbb{V}$ la variance d'une variable aléatoire (on suppose ici $f^* \equiv \tilde{f}^*$). On peut alors écrire [DMS+08] :

$$\begin{aligned}
Err^2(\mathbf{x}) &= \mathbb{E}_{y|\mathbf{x}}(\ell(f^*(\mathbf{x}), y)) = \mathbb{E}_{y|\mathbf{x}}([y - f^*(\mathbf{x})]^2) \\
Err^2(\mathbf{x}) &= \mathbb{V}_{y|\mathbf{x}}([y - f^*(\mathbf{x})]) + \left(\mathbb{E}_{y|\mathbf{x}}([y - f^*(\mathbf{x})])\right)^2 \\
Err^2(\mathbf{x}) &= \mathbb{V}_{y|\mathbf{x}}([y - f_{vrai}(\mathbf{x}) + f_{vrai}(\mathbf{x}) - f^*(\mathbf{x})]) + \left(\mathbb{E}_{y|\mathbf{x}}([y - f_{vrai}(\mathbf{x}) + f_{vrai}(\mathbf{x}) - f^*(\mathbf{x})])\right)^2 \\
Err^2(\mathbf{x}) &= \mathbb{V}_{y|\mathbf{x}}([\varepsilon + f_{vrai}(\mathbf{x}) - f^*(\mathbf{x})]) + \left(\mathbb{E}_{y|\mathbf{x}}([\varepsilon + f_{vrai} - f^*(\mathbf{x})])\right)^2
\end{aligned}$$

Or, $f_{vrai}(\mathbf{x})$ est fixe donc sa variance est nulle. De plus, ε est d'espérance nulle et indépendant du modèle appris.

On peut donc simplifier l'expression sous la forme :

$$Err^2(x) = \sigma^2 + \mathbb{V}_{y|\mathbf{x}}(f^*(\mathbf{x})) + \left(\mathbb{E}_{y|\mathbf{x}}([f_{vrai} - f^*(\mathbf{x})])\right)^2 \tag{3.8}$$

Cette expression fait apparaître les 3 sources d'erreur possibles lors d'un apprentissage statistique. Tout d'abord, le bruit sur les données et le premier terme illustre clairement que l'on ne peut pas s'attendre à une erreur plus faible que l'ordre de grandeur de la variance du bruit. Le deuxième terme représente la variance de la prédiction du modèle $f^*(\mathbf{x})$. Il est lié à la flexibilité de la famille de fonction choisie et donc, d'une certaine façon, au nombre de paramètres libres, ou degrés de liberté du modèle. Enfin, le dernier terme représente le biais du modèle et donc la distance entre le modèle réel et le meilleur modèle possible dans la famille de fonctions choisie. La complexité d'un modèle (mesurée souvent en nombre de paramètres ou degrés effectifs de liberté [HTF09]) doit donc toujours résulter d'un compromis entre une complexité trop faible qui impliquerait un biais important et une complexité trop importante qui impliquerait une sensibilité trop forte aux données d'apprentissage.

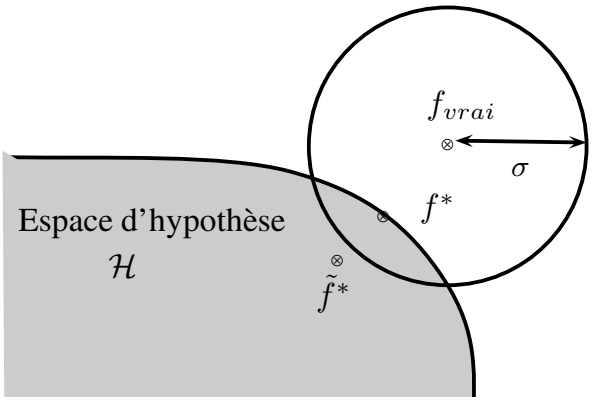


FIGURE 3.2 : Vue conceptuelle des différents types d'erreurs commises lors d'un apprentissage statistique. Même si la famille de fonctions choisie pour contenir la solution est adaptée au problème et que l'algorithme d'apprentissage fonctionne bien, l'écart avec le vrai modèle des données peut être très important du fait des autres sources d'erreurs possibles.

3.2.2.3 La régularisation

Comme nous l'avions souligné, une alternative à la minimisation du risque empirique consiste à imposer des contraintes supplémentaires sur le problème. Une approche simple est la régularisation [Wah90, GJP95]. L'idée est donc de résoudre le problème régularisé par un schéma de type Tikhonov [TA77]

$$f^* = \underset{f \in \mathscr{H}}{\operatorname{argmin}} R_e(f) + \lambda \Omega(f) \tag{3.9}$$

avec $\Omega(f)$ une fonction régularisante définie par

$$\begin{array}{rccc} \Omega : & \mathscr{H} & \longrightarrow & \mathbb{R}^+ \\ & f & \longmapsto & \Omega(f). \end{array}$$

Le paramètre de régularisation $\lambda \geq 0$ règle le compromis entre la minimisation du risque empirique et la complexité du modèle. D'autres méthodes de régularisation peuvent toutefois être condidérées pour résoudre ce type de problème, notamment :

– la régularisation de type Ivanov [Iva76] qui conduit au problème

$$\begin{array}{rl} f^* = & \operatorname{argmin}_{f \in \mathscr{H}} R_e(f) \\ & \text{s.t.} \quad \Omega(f) \leq \lambda, \end{array}$$

– la régularisation de type Morozov [Mor84] conduisant à

$$\begin{aligned} f^* = \quad & \operatorname{argmin}_{f\in\mathscr{H}} \Omega(f) \\ & \text{s.t.} \quad R_e(f) \leq \lambda . \end{aligned}$$

Sous des conditions essentiellement de convexité des fonctions $R_e(f)$ et $\Omega(f)$, on peut établir que les trois types de régularisation pré-cités conduisent à des solutions identiques [Mie99, pages 12-13].

Les fonctions de régularisations les plus utilisées sont :

– La norme quadratique fonctionnelle
  En supposant que la fonction de décision f est issue d'un espace d'hypothèses $\mathscr{H}$ muni du produit scalaire $\langle f,g\rangle_{\mathscr{H}}$ et que ce produit scalaire induit une norme $\|f\|_{\mathscr{H}}$, on définit alors :

$$\Omega(f) = \|f\|^2_{\mathscr{H}} = \langle f,f\rangle_{\mathscr{H}}.$$

  Ce type de régularisation est souvent utilisé dans le cadre des méthodes à noyaux [SS01].
– La norme ℓ_2 sur les paramètres du modèle
  Une autre forme de régularisation quadratique est donnée par

$$\Omega(f) = \|\mathbf{w}\|_2^2 = \sum_j w_j^2$$

  si on exprime la fonction de décision comme issue d'une famille génératrice (voir équation 3.6). w_j désigne un coefficient réel de l'expansion sur la base de fonctions.
– La norme ℓ_1 sur les paramètres du modèle

$$\Omega(f) = \|\mathbf{w}\|_1 = \sum_j |w_j|$$

  Cette régularisation a la propriété d'introduire la parcimonie de la fonction f et privilégie donc les fonctions à faible nombre de paramètres.

Les deux premiers types de régularisation (régularisation quadratique) favorisent des fonctions d'estimation lisses alors que le dernier favorise la parcimonie. Précisons que cette parcimonie peut être renforcée en considérant des fonctions régularisantes non-convexes comme la pseudo-norme ℓ_p, $p<1$ définie par $\Omega(f) = \sum_j |w_j|^p$. Une contrainte plus forte sur la régularité de la fonction f peut être imposée en considérant une norme $\Omega(f) = \|\mathbf{w}\|_q$, $q>2$. Soulignons aussi que certaines méthodes d'arrêt prématuré de l'optimisation lors de la phase d'apprentissage (appelées *early-stopping*) peuvent également être rapprochées de la régularisation. En effet, il a été établi que pour les techniques d'apprentissage en ligne (utilisées par exemple pour les réseaux de neurones), cet arrêt prématuré favorise la bonne généralisation du modèle [Pre98].

3.2.3 La sélection de modèle

En pratique, les familles de fonctions possèdent elles aussi le plus souvent des hyper-paramètres qui doivent être optimisés. Ainsi pour les réseaux de neurones, il faut déterminer le nombre de couches cachées et le nombre d'unités par couches. Les méthodes à noyaux nécessitent quant à elles le choix d'un noyau et la détermination de ses hyper-paramètres. Pour permettre d'obtenir également les valeurs optimales sur ces hyper-paramètres, il est alors nécessaire de définir une étape supplémentaire dite de sélection de modèle. En général, la sélection du modèle le plus performant s'effectue à partir d'une estimation de l'erreur de généralisation.

Plusieurs méthodes sont disponibles pour mener à bien cette étape [HTF09]. La méthode la plus classique consiste à utiliser une partie des données, appelée ensemble de validation, $\left\{(\mathbf{x}'_j,y'_j)\right\}_{j=1,\cdots,N'}$ qui sera réservée pour cette étape. L'erreur de généralisation $R(f)$ est alors estimée par le risque empirique évalué sur ce nouvel ensemble :

$$R(f) = R'_e(f) = \frac{1}{N'}\sum_{j=1}^{N'} \ell(f(\mathbf{x}_j)',y'_j) \tag{3.10}$$

Cette approche est possible lorsque le nombre d'échantillons disponibles est très large. Lorsque ce nombre est plus limité, un ensemble de validation de taille trop faible risque de faire ressurgir les risques de sur-apprentissage. L'idée de la validation croisée est alors de fusionner les ensembles d'apprentissage et de validation. Ce nouvel ensemble est ensuite divisé en un nombre fixé K de sous-ensembles. Un sous-ensemble k est alors choisi pour la validation et les ensembles restants sont utilisés pour l'apprentissage. La procédure est ensuite itérée pour chaque sous-ensemble et l'on moyenne les risques empiriques ainsi obtenus. Ainsi, on a :

$$R(f) = \frac{1}{K}\sum_{k=1}^{K} R_{e_k}(f) \tag{3.11}$$

où R_{e_k} représente le risque empirique évalué sur le $k^{\text{ème}}$ sous-ensemble de données. Lorsque la validation croisée est poussée à l'extrême avec des sous-ensembles de taille 1, elle porte le nom de "*leave-one-out validation*" [Wah90].

Une autre façon de procéder à la sélection de modèle consiste à estimer l'erreur de généralisation à partir de l'erreur empirique. Pour ce faire, on considère une famille (finie ou dénombrable) de fonctions $\mathscr{H}_d$ à complexité (nombre de paramètres) d croissante puis on sélectionne le modèle minimisant le critère composite (ou critère pénalisé)

$$R_e(f) + \text{pen}(d,N) \tag{3.12}$$

où N désigne le nombre de données disponibles et où le terme de pénalisation $\text{pen}(d,N)$ permet de donner plus d'importance à des modèles précis avec un faible nombre de paramètres. Cette formulation, souvent appelée « *score* », diffère de l'approche régularisation par l'absence de paramètre de régularisation à fixer. Les différentes approches de sélection de modèle basées sur (3.12) diffèrent par le choix du terme de pénalisation. Pour illustrer ces différentes pénalisations, on se place dans le cas d'un modèle linéaire, où il est possbile d'obtenir une formule exacte pour chaque cas :

$$y = \mathbf{X}\mathbf{w} + \zeta \tag{3.13}$$

où y corresponds au vecteur de sortie, $\mathbf{X}$ est la matrice des entrées, $\mathbf{w}$ le vecteur des paramètres du modèle et ζ le vecteur des erreurs du modèle. On considère de plus R_e le risque empirique du modèle pour une fonction coût quadratique et σ^2 la variance estimée de l'erreur obtenue en utilisant le modèle le plus complet possible. On obtient alors les expressions des différents scores [HY03] qui sont décrites dans le tableau 3.2.

Nom	$\text{pen}(d,N)$
AIC [Aka73]	$N\log(R_e) + 2d$
BIC [Sch78]	$N\log(R_e) + d log(N)$
Cp de Mallows [Mal]	$NR_e/\sigma^2 + 2d - N$
MDL [Ris78]	$(N-d)\log(R_e) + k\log(\mathbf{w}^\top\mathbf{X}^\top\mathbf{X}\mathbf{w}) + (N-d-1)\log(N/(N-d)) - (d+1)\log(d)$

TABLE 3.2 : Présentation de quelques critères de sélection de modèle avec les termes de pénalisation associés.

Dans la pratique, le problème posé par cette approche de généralisation est la spécification de la complexité du modèle. Dans certains cas (comme des algorithmes d'optimisation de type régression ridge [Wah90]), cette complexité est reliée à la notion de nombre effectif de paramètres ou de degrés de liberté [Wah90, HTF09]. Néanmoins, l'approche validation est souvent préférée dans la pratique notamment si l'on dispose d'un nombre suffisant de données pour mettre en oeuvre la validation croisée.

3.3 Modèles dynamiques

La modélisation d'un système dynamique diffère à certains égards du cadre général décrit précédemment car les données sont ordonnées temporellement et l'on présuppose que cet ordre a une importance dans la modélisation. Ainsi, si l'on considère à l'instant t un système possédant une seule entrée $u \in \mathscr{U}$ et une seule sortie $y \in \mathscr{Y}$, on obtient une observation $(u(t), y(t))$. Dans ce document, nous n'étudierons que des modèles à temps discret où les

Modèle	Acronyme	Entrées $\mathbf{U}_{k-n_u\|k-1}$	Sorties $\mathbf{Y}_{k-n_y\|k-1}$	Prédictions $\hat{\mathbf{Y}}_{k-n_{\hat{y}}\|k-1}$
Finite Impulse Response	FIR	x		
AutoRegressive with eXogenous inputs	ARX	x	x	
AutoRegressive Moving Average with eXogenous inputs	ARMAX	x	x	x
Output Error	OE	x		x

TABLE 3.3 : Vecteur de régression utilisé en fonction des différents modèles dynamiques.

observations sont prises à un intervalle de temps régulier. Si cet intervalle noté T est défini (T représente la période d'échantillonnage), alors la variable temporelle t peut être mise sous la forme $t = kT$ avec $k \in \mathbb{N}$. On peut alors utiliser la notation suivante pour les entrées/sorties $u(t) = u(kT) = u_k$.

Le but d'un modèle dynamique est d'obtenir une prédiction $\hat{y}_k$ de la sortie y_k connaissant les observations passées (on ne considère que des modèles causaux), regroupées dans les vecteurs

$$\mathbf{U}_{0|k-1} = \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_i \\ \vdots \\ u_{k-1} \end{bmatrix} \in \mathscr{U}^k \quad \text{et} \quad \mathbf{Y}_{0|k-1} = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_i \\ \vdots \\ y_{k-1} \end{bmatrix} \in \mathscr{Y}^k. \tag{3.14}$$

$$\hat{y}_k = f(\mathbf{U}_{0|k-1}, \mathbf{Y}_{0|k-1}) \tag{3.15}$$

Toutefois, les vecteurs $\mathbf{U}_{0|k-1}$ et $\mathbf{Y}_{0|k-1}$ croissent à chaque instant. Or, le modèle recherché est de taille fixe. Il faut donc définir des « fenêtres temporelles », de taille respectivement n_u pour l'entrée et n_y pour la sortie, qui limitent les dimensions de ces 2 vecteurs. Ces paramètres permettent de définir le « vecteur de régression » φ_k du modèle à l'instant kT de la façon suivante :

$$\varphi_k = \begin{bmatrix} \mathbf{U}_{k-n_u|k-1}^\top & \mathbf{Y}_{k-n_y|k-1}^\top \end{bmatrix}^\top$$

Ce vecteur de régression peut également être alimenté par les prédictions données par le modèle aux instants précédents $\hat{\mathbf{Y}}_{k-n_{\hat{y}}|k-1}$ et donc par l'erreur commise par le modèle, définie par $\mathbf{E}_{k-n_e|k-1} = \mathbf{Y}_{k-n_e|k-1} - \hat{\mathbf{Y}}_{k-n_e|k-1}$, où n_e représente la taille de la fenêtre temporelle retenue pour l'erreur du modèle. Ces différents choix possibles de φ_k aboutissent à différents modèles définis selon le vecteur de régression utilisé et sont présentés dans le tableau 3.3. D'autres combinaisons sont possibles et sont décrites dans le livre de L. Ljung [Lju02] pour un état des lieux complet.

Dans la suite de ce chapitre, nous allons nous focaliser sur différents types de modélisation dynamique dérivés de ces différents vecteurs de régression. Dans un premier temps, nous allons considérer une modélisation linéaire puis nous allons étendre notre étude à des modèles de types réseaux de neurones ou méthodes à noyaux. Finalement, nous aborderons le cas des réseaux bayésiens dynamiques. À chaque type de modélisation, nous préciserons les problématiques d'identification des modèles, l'application aux données thermiques (bases des données « Transistor » et « Maquette ») et l'analyse des résultats expérimentaux.

3.4 Modèles linéaires

Les modèle linéaires (ou pseudo-linéaires dans le cas OE même si cette distinction ne sera plus précisée par la suite) sont définis par la formule :

$$\hat{y}_k = \boldsymbol{\varphi}_k^\top \mathbf{w}. \tag{3.16}$$

Pour comparer les différents modèles linéaires, il est utile d'étudier leurs fonctions de transfert. Pour cela, on introduit l'opérateur « retard » q^{-1} tel que $q^{-1}u_k = u_{k-1}$. Si l'on ne définit pas de fenêtre temporelle, un système linéaire à temps discret peut alors se mettre sous la forme :

$$y_k = G(q)u_k + H(q)\zeta_k$$

avec

$$G(q) = \sum_{k=1}^{\infty} g_k q^{-k} \quad \text{et} \quad H(q) = 1 + \sum_{k=1}^{\infty} h_k q^{-k}$$

où ζ_k représente alors les perturbations qui peuvent être liées à un bruit de mesure ou à des entrées incontrôlables. C'est sous cette forme que nous allons présenter les trois types de modèles linéaires qui seront étudiés dans ce chapitre. Pour des raisons de simplicité, nous présenterons la problématique d'identification d'un modèle linéaire pour un système mono-entrée mono-sortie, l'extension au cas multi-entrée se fait de façon directe.

3.4.1 Les différents types de modèles linéaires considérés

3.4.1.1 Modèles à erreur d'équation

Le plus simple des modèles linéaires est le FIR ou filtre à réponse impulsionnelle finie puisque la sortie du modèle est uniquement déterminée par ses entrées. La prédiction fournie par le modèle FIR à l'instant kT, $k = n_u, \cdots, N$ est

$$\hat{y}_k = \boldsymbol{\varphi}_k^\top \mathbf{w} \quad \text{avec} \quad \boldsymbol{\varphi}_k = \mathbf{U}_{k-n_u|k-1}. \tag{3.17}$$

Le modèle recherché est donc représenté par l'équation :

$$y_k = b_1 u_{k-1} + \cdots + b_{n_u} u_{k-n_u} + \zeta_k \tag{3.18}$$

On peut alors voir que :

$$G(q) = \sum_{k=1}^{n_u} b_k q^{-k} = B(q) \quad \text{et} \quad H(q) = 1.$$

Si l'on intègre également les mesures passées de la sortie au vecteur de régression, on obtient alors un modèle de type ARX, modèle autorégressif à entrée exogène, où l'on a :

$$\hat{y}_k = \boldsymbol{\varphi}_k^\top \mathbf{w} \quad \text{avec} \quad \boldsymbol{\varphi}_k = \begin{bmatrix} \mathbf{U}_{k-n_u|k-1}^\top & -\mathbf{Y}_{k-n_y|k-1}^\top \end{bmatrix}^\top. \tag{3.19}$$

L'équation du modèle est donc :

$$y_k = -a_1 y_{k-1} - \cdots - a_{n_y} y_{k-n_y} + b_1 u_{k-1} + \cdots + b_{n_u} u_{k-n_u} + \zeta_k. \tag{3.20}$$

On peut alors écrire :

$$G(q) = \frac{\sum_{k=1}^{n_u} b_k q^{-k}}{1 + \sum_{k=1}^{n_y} a_k q^{-k}} = \frac{B(q)}{A(q)} \quad \text{et} \quad H(q) = \frac{1}{A(q)}$$

avec $A(q)$ et $B(q)$ des polynômes en q^{-1} dont les coefficients sont respectivement les paramètres a_j et b_j.

3.4.1.2 Modèle à erreur de sortie

Lorsque les sorties mesurées passées ne peuvent pas être intégrées au vecteur de régression (si elles sont notamment indisponibles au moment de l'utilisation du modèle comme dans le cas de notre application sur l'estimation de la température des composants hyperfréquence) et que l'on suppose que les perturbations du système ne sont liées qu'à un bruit de mesure, il est possible d'utiliser un modèle OE, dit à erreur de sortie. On a alors :

$$\hat{y}_k = \varphi_k^\top \mathbf{w} \quad \text{avec} \quad \varphi_k = \begin{bmatrix} \mathbf{U}_{k-n_u|k-1}^\top & -\hat{\mathbf{Y}}_{k-n_{\hat{y}}|k-1}^\top \end{bmatrix}^\top. \tag{3.21}$$

Le modèle est ainsi représenté par l'équation :

$$\hat{y}_k = -f_1\hat{y}_{k-1} + \cdots - f_{n_{\hat{y}}}\hat{y}_{k-n_{\hat{y}}} + b_1 u_{k-1} + \cdots + b_{n_u} u_{k-n_u}. \tag{3.22}$$

Par conséquent, on suppose que la sortie mesurée y_k est donnée par l'expression :

$$y_k = \hat{y}_k + \zeta_k.$$

et on en déduit que :

$$G(q) = \frac{\sum_{k=1}^{n_u} b_k q^{-k}}{1 + \sum_{k=1}^{n_y} f_k q^{-k}} = \frac{B(q)}{F(q)} \quad \text{et} \quad H(q) = 1.$$

3.4.2 Identification des paramètres d'un modèle linéaire

Pour les modèles à erreur d'équation, l'identification des paramètres est aisée. En effet, la totalité des données nécessaires est disponible et le modèle est linéaire. On part de l'équation générale :

$$\hat{y}_k = \varphi_k^\top \mathbf{w}$$

avec le vecteur de régression φ_k défini respectivement par l'équation (3.17) pour le modèle FIR et (3.19) pour l'ARX. Les éléments de $\mathbf{w}$ correspondants sont issus des équations (3.18) et (3.20) pour les modèles FIR et ARX respectivement. Pour ce type de problème, la fonction de coût choisie est le plus souvent quadratique ; le problème devient alors un problème d'estimation au sens « *des moindres carrés* ». En se référant aux expressions des vecteurs de régression, on a $n = n_u$ pour le FIR et $n = \max(n_y, n_u)$ pour le modèle ARX. Le risque empirique peut alors se mettre sous la forme :

$$R_e(\mathbf{w}) = \frac{1}{N-n+1}\mathbf{E}_{n|N}^\top \mathbf{E}_{n|N} = \frac{1}{N-n+1}(\mathbf{Y}_{n|N} - \hat{\mathbf{Y}}_{n|N})^\top (\mathbf{Y}_{n|N} - \hat{\mathbf{Y}}_{n|N})$$

où n représente le décalage temporel nécessaire pour alimenter le vecteur de régression du modèle considéré. Soit ψ, la matrice de régression

$$\psi = \begin{bmatrix} \varphi_n^\top \\ \varphi_{n+1}^\top \\ \vdots \\ \varphi_k^\top \\ \vdots \\ \varphi_N^\top \end{bmatrix}.$$

On obtient alors cette expression du vecteur des prédictions du modèle

$$\hat{\mathbf{Y}}_{n|N} = \psi \mathbf{w}$$

à partir de laquelle on reformule le risque empirique comme

$$R_e(\mathbf{w}) = \frac{1}{N-n+1}(\mathbf{Y}_{n|N} - \psi\mathbf{w})^\top (\mathbf{Y}_{n|N} - \psi\mathbf{w}).$$

La condition d'optimalité de ce problème par rapport au vecteur de paramètres $\mathbf{w}$

$$\nabla_{\mathbf{w}} R_e = 0 \quad \text{avec} \quad \nabla_{\mathbf{w}} R_e = -\frac{2}{N-n+1} \psi^\top \left(\mathbf{Y}_{n|N} - \psi \mathbf{w} \right)$$

fournit la solution

$$\mathbf{w}^* = (\psi^\top \psi)^{-1} \psi^\top \mathbf{Y}_{n|N} \tag{3.23}$$

L'estimation des paramètres des modèles à erreur de sortie présente quelques difficultés supplémentaires. En effet, on constate que le vecteur de régression (3.21) dépend des paramètres du modèle puisqu'il fait intervenir les versions décalées temporellement de la sortie $\hat{y}_k$. Le problème d'optimisation n'est donc pas soluble de manière analytique et il faut utiliser des méthodes numériques de type descente de gradient pour obtenir les paramètres du modèle [NW06]. D'une façon générique, ces méthodes procèdent par minimisation de la fonction du risque selon le schéma itératif

$$\mathbf{w}_{t+1} = \mathbf{w}_t + \gamma_t \mathbf{d}_t \tag{3.24}$$

où $\mathbf{d}_t$ représente la direction de recherche à l'itération t et γ_t le pas de recherche de façon à assurer la décroissance du critère $R_e(\mathbf{w})$. Il est bien connu que la direction de recherche a généralement la forme suivante

$$\mathbf{d}_t = -\mathbf{H}_t^{-1} \nabla_{\mathbf{w}} R_e(\mathbf{w}_t)$$

où $\nabla_{\mathbf{w}} R_e(\mathbf{w}_t)$ est le gradient du critère par rapport à $\mathbf{w}$ évalué à partir de la solution courante $\mathbf{w}_t$ et $\mathbf{H}_t$ une matrice symétrique et non-singulière. La structure de la matrice $\mathbf{H}$ varie en fonction du type d'algorithme utilisé. Ainsi, on distingue :

- la méthode de descente du gradient pour laquelle on a $\mathbf{H} = \mathbf{I}$ avec $\mathbf{I}$ la matrice identité,
- la méthode de Newton où $\mathbf{H}$ est la matrice hessienne $\nabla_{\mathbf{w}\mathbf{w}^\top} R_e(\mathbf{w})$,
- les méthodes de type quasi-Newton qui considèrent $\mathbf{H}$ comme une approximation de la matrice hessienne dérivée à partir de la connaissance du gradient.

La procédure itérative (3.24) est répétée jusqu'à la convergence de l'algorithme d'optimisation vers au moins un minimum local. Le lecteur intéressé par les éléments peut notamment se référer à l'ouvrage de Nocedal et Wrigth [NW06].

L'application de cette procédure à l'estimation des paramètres d'un modèle OE nécessite au minimum la détermination du gradient dont nous allons expliciter ci-dessous les calculs. Pour ce faire, ré-écrivons le critère sous la forme :

$$R_e(\mathbf{w}) = \frac{1}{N-n} \sum_{k=n}^{N} (y_k - \hat{y}_k)^2 \quad \text{avec} \quad n = \max(n_u, n_{\hat{y}}).$$

L'expression du gradient recherché est alors :

$$\nabla_{\mathbf{w}} R_e(\mathbf{w}) = -\frac{2}{N-n} \sum_{k=n}^{N} (y_k - \hat{y}_k) \nabla_{\mathbf{w}} \hat{y}_k$$

et elle devient complète si l'on dispose de l'expression de $\nabla_{\mathbf{w}} \hat{y}_k$. Or, pour un modèle OE, on sait que le vecteur de paramètres est $\mathbf{w} = \left[b_1 \cdots b_{n_u} - f_1 \cdots - f_{n_{\hat{y}}} \right]^\top$ et la prédiction est donnée par $\hat{y}_k = \varphi_k^\top \mathbf{w}$ comme à l'équation (3.21). On peut alors écrire :

$$\begin{aligned}
\nabla_{f_j} \hat{y}_k &= -f_1 \nabla_{f_j} \hat{y}_{k-1} - \cdots - f_{n_{\hat{y}}} \nabla_{f_j} \hat{y}_{k-n_{\hat{y}}} - \hat{y}_{k-j}, \qquad \forall j = 1, \cdots, n_{\hat{y}} \\
\nabla_{b_j} \hat{y}_k &= -f_1 \nabla_{b_j} \hat{y}_{k-1} - \cdots - f_{n_{\hat{y}}} \nabla_{b_j} \hat{y}_{k-n_{\hat{y}}} + u_{k-j}, \qquad \forall j = 1, \cdots, n_u
\end{aligned}$$

En combinant cette série d'équations, on arrive à écrire le gradient sous la forme compacte

$$\nabla_{\mathbf{w}} \hat{y}_k = \frac{1}{F(q)} \varphi_k$$

avec $F(q)$ est tel que défini dans la sous-section 3.4.1.2. On constate que la fonction de sensibilité par rapport aux paramètres $\mathbf{w}$ de la sortie du modèle à l'instant k est obtenue par filtrage des gradients précédents.

En pratique, dans la boite à outils [Lju00] que nous avons utilisée, la méthode de Levenberg-Marquardt basée sur l'approximation suivante de la hessienne

$$\mathbf{H} = \frac{2}{N-n}\sum_{k=n}^{N} \nabla_{\mathbf{w}}\hat{y}_k \nabla_{\mathbf{w}}\hat{y}_k^{\top}$$

est utilisée. L'algorithme peut être initialisé à partir d'un modèle ARX avec $n_y = n_{\hat{y}}$ et la même taille de fenêtre temporelle n_u sur l'entrée. Une autre approche d'identification des paramètres du modèle OE utilisée dans la boîte à outil repose sur les méthodes de sous-espace [Lju02] dont les détails seront exposés dans le chapitre suivant.

3.4.3 Problématique d'identification liée à ces modèles linéaires

Une fois le type de modèle linéaire et la procédure d'identification des paramètres fixés, le seul élément à déterminer reste l'ordre (nombre de retards sur l'entrée et la sortie i.e. n_u, n_y ou $n_{\hat{y}}$ selon les types de modèles) du vecteur de régression. Ce choix se ramène à une sélection de modèle telle que décrite précédemment. Une procédure de validation croisée est donc le plus souvent utilisée pour cette étape afin d'obtenir une évaluation de l'erreur de généralisation pour les différents ordres choisis.

Pour valider les différents modèles, nous utilisons le critère quadratique moyen suivant

$$\mathrm{J_{val}} = \frac{1}{N'}\sum_{k=n}^{N'} (y_k - \hat{y}_k)^2 \qquad \text{avec} \tag{3.25a}$$

$$\hat{y}_k = -\sum_{j=1}^{n_\theta} \theta_j \hat{y}_{k-j} + \sum_{j=1}^{n_u} b_j u_{k-j} \tag{3.25b}$$

avec N' la taille du jeu de validation. Pour un modèle FIR, on a les coefficients $\theta_j = 0$, $\forall j$ dans (3.25b) alors que pour un modèle OE, on a $\theta_j = f_j$ et $n_\theta = n_{\hat{y}}$ conformément à l'expression d'un modèle OE (3.22). En revanche lorsqu'il s'agira d'un modèle ARX, bien que ses paramètres aient été estimés à partir des vraies mesures y, le calcul de $\mathrm{J_{val}}$ sera basé sur la sortie simulée du modèle et ne fait pas appel aux mesures réelles. On choisira pour ce faire $\theta_j = a_j$ avec $n_\theta = n_y$.

3.4.4 Tests sur les bases de données thermiques

3.4.4.1 Présentation des données utilisées et protocole expérimental

On choisit de comparer les modèles FIR, OE et ARX dans le cas de la simulation (la sortie réelle est supposée inconnue), ce qui correspond au fait que les instruments de mesure de température ne pourront pas être utilisés durant le fonctionnement normal du système RADAR. Pour le modèle ARX, ce sont donc les sorties prédites aux instants précédents qui sont réinjectées en entrée du modèle pour la prédiction à l'instant t (contrairement à la définition théorique de ce modèle qui utilise la vraie sortie du système). Les fenêtres temporelles testées sont de 1 à 15 retards pour les entrées et de 0 (modèle FIR) à 3 retards pour la sortie.

Pour les données de la base « Transistor », on dispose d'une base de 10000 points [1]. La base de données est donc divisée en 2 parties égales de 5000, l'une étant utilisée pour l'apprentissage et la sélection des modèles, l'autre afin de déterminer les performances en test du modèle retenu. Pour la sélection de modèles, la validation croisée doit se faire sans permutation aléatoire des données pour tenir compte de l'aspect temporel des données. On génère

1. Pour rappel, la base « Transistor » est issue des simulations à éléments finis. L'entrée du modèle est la puissance injectée et la sortie est la température de jonction du composant.

donc 10 parties de 1000 échantillons par décalage successif de 400 échantillons dans la base d'apprentissage. Lors de l'utilisation de ces données, il est apparu que l'utilisation d'une de ces parties était suffisante pour obtenir un apprentissage satisfaisant. Pour la validation croisée, un seul sous-échantillon est donc utilisé pour l'apprentissage et l'on mesure l'erreur quadratique obtenue sur les parties restantes. Pour calculer l'erreur de test des modèles retenus, on apprend le modèle sur l'intégralité des données utilisées pour la sélection de modèle et l'on mesure l'erreur obtenue sur les données réservées pour cette étape. De plus, pour ces données, on dispose également des données sans bruit. On comparera donc également les modèles avec ces données. Précisons enfin que les performances en test sont évaluées de la même manière que J_{val} (équation 3.25).

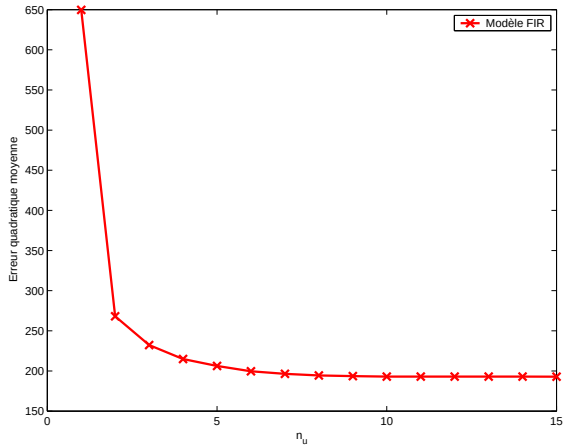
Pour les données de la base « Maquette » [2], on dispose de cinq séries de mesures de 2400 points. Une de ces séries est réservée pour évaluer les performances en test des différents modèles sélectionnés. Pour la phase d'apprentissage et de validation des hyper-paramètres, il reste donc 4 bases. Ces quatre bases restantes sont donc utilisées pour réaliser une validation croisée. À chaque itération,trois séries de mesures sont utilisées pour l'apprentissage et une pour calculer les performances du modèle. Des transitions sont insérées entre les données d'apprentissage afin de créer une seule base d'apprentissage à chaque étape de la validation croisée. Ces transitions sont réalisées en supposant une puissance injectée nulle et une ventilation maximale, ce qui représente le cas où le retour à la température ambiante est le plus rapide possible.

3.4.4.2 Résultats obtenus

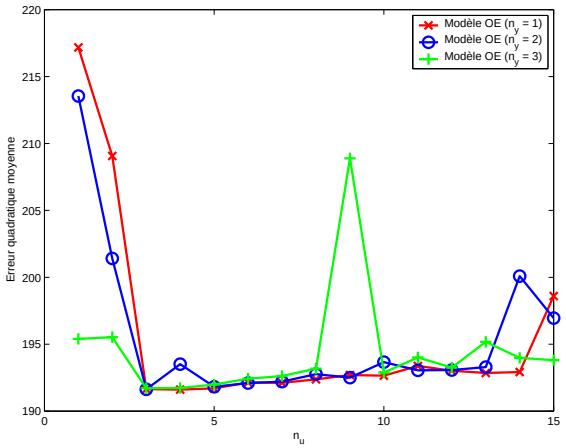
Pour la base de données « Transistor », l'erreur quadratique moyenne sur les ensembles de validation des différents modèles FIR (figure 3.3(a)), OE (figure 3.3(b)) et ARX (figure 3.3(c)) sont présentés. L'erreur la plus faible est obtenue par un modèle OE. Toutefois, on constate que les performances optimales des modèles ARX et OE sont relativement proches. Le modèle FIR n'atteint ces performances que pour des fenêtres temporelles plus longues ($n_u > 15$). Pour les modèles ARX, l'évolution de l'erreur est relativement logique, elle décroit avec l'augmentation de la taille des fenêtres temporelles sur la sortie ou sur l'entrée. Pour les modèles OE, on constate que les performances se dégradent pour des fenêtres temporelles telles que $n_u > 5$. En particulier, pour $n_{\hat{y}} = 3$ et $n_u = 9$, on remarque que l'erreur de validation augmente fortement. Ceci pourrait s'expliquer par le fait que l'un des modèles OE identifiés pour ces paramètres est faiblement stable. De plus, des performances satisfaisantes semblent atteintes avec un seul retard sur la sortie, ce qui n'est pas le cas pour les modèles ARX avec de faibles retards sur l'entrée. Les résultats optimaux de chaque modèle sont présentés dans la table 3.4. Une partie des réponses sur la base de test est présentée sur la figure 3.3(d). Au final, le modèle retenu est un modèle OE avec une fenêtre temporelle de 1 sur la sortie et de 4 sur l'entrée. Ces paramètres seront par la suite utilisés pour les modélisations non-linéaires (réseaux de neurones et support vector machines). On constate également que les erreurs sont plus faibles vis-à-vis des données sans bruit. Cela laisse supposer que les modèles sont relativement proches du modèle qui a servi à générer les données. Ceci est confirmé par le tracé de l'erreur absolue du modèle OE (figure 3.4). La forte périodicité de l'erreur ne peut venir ni des données d'entrée qui ont été générées aléatoirement selon un schéma SBPA, ni du bruit ajouté, ni du modèle dont la fenêtre temporelle est bien plus courte. Cette erreur semble donc être liée au schéma numérique utilisé par le modèle à éléments finis (pas d'échantillonnage légèrement inconstant). Les tests sur l'autocorrélation de l'erreur du modèle OE sur ces données sans bruit (figure 3.5) semblent confirmer que le modèle a appris correctement les données même si les valeurs dépassent légèrement les seuils de confiance à 95%.

Pour la base de données « Maquette », le retard sur les 2 entrées est fixé à la même valeur. Les erreurs quadratiques moyennes sur les ensembles de validation des différents modèles FIR (figure 3.6(a)) et OE (figure 3.6(b)) sont présentées. Les résultats des modèles ARX ne peuvent pas être présentés car la plupart des modèles identifiés se sont révélés instables. Sur les modèles FIR, l'erreur décroit logiquement avec le nombre de retards sur l'entrée. Pour les modèles OE, avec 2 ou 3 retards sur la sortie, les résultats sont très irréguliers et certains modèles identifiés

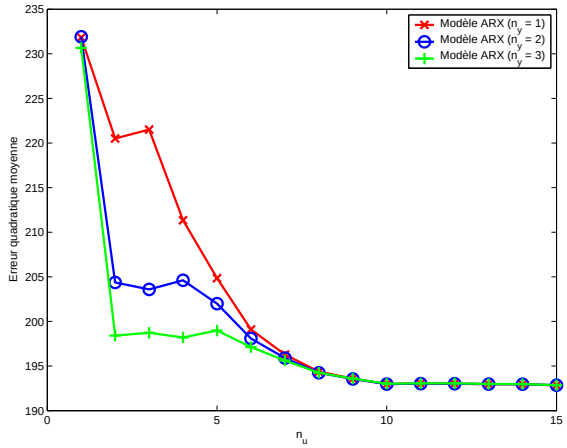
2. Les données « Maquette » sont celles relevées sur la maquette expérimentale. Elles comprennent deux entrées, la puissance injectée et la vitesse de l'air dans le système de refroidissement, et une sortie, la température sous le composant.



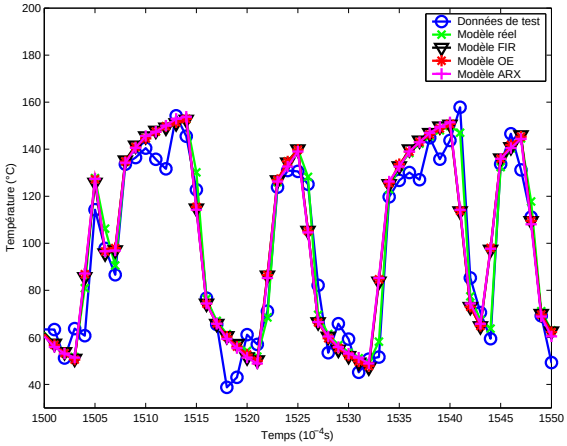
(a) Courbe du critère de validation obtenue pour une modélisation FIR linéaire.



(b) Courbes du critère de validation obtenues pour un modèle OE linéaire.



(c) Courbes du critère de validation obtenues pour un modèle ARX linéaire testé en simulation.



(d) Comparaison des sorties des modèles linéaire optimaux sur une partie des données de test. Précisons que « Modèle réel » représente les données non bruitées (voir chapitre 1).

FIGURE 3.3 : Résumé graphique des résultats obtenus par les méthodes d'identification linéaire sur la base de données « Transistor ».

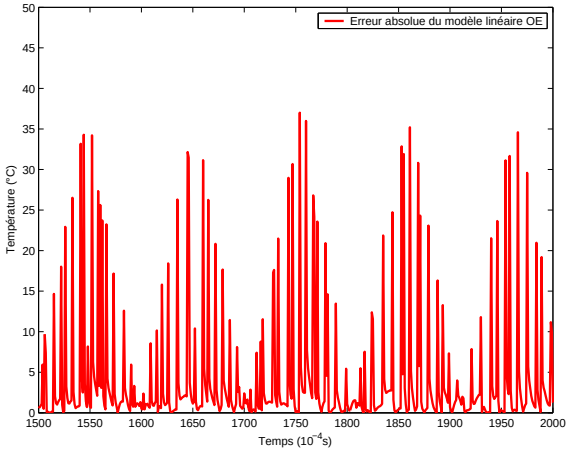


FIGURE 3.4 : Erreurs absolues entre le modèle linéaire OE et les données « Transistor » sans bruit.

Type de Modèle	Ordres optimaux	Erreur de validation	Erreur de test	Erreur de test (données sans bruit)
FIR	$n_u = 15$	192.833	186.104	80.576
ARX	$n_u = 10$ et $n_y = 1$	192.989	187.470	82.453
OE	$n_u = 4$ et $n_y = 1$	191.610	185.831	80.184

TABLE 3.4 : Récapitulatif des résultats fournis par les modèles linéaires optimaux sur la base de données « Transistor ».

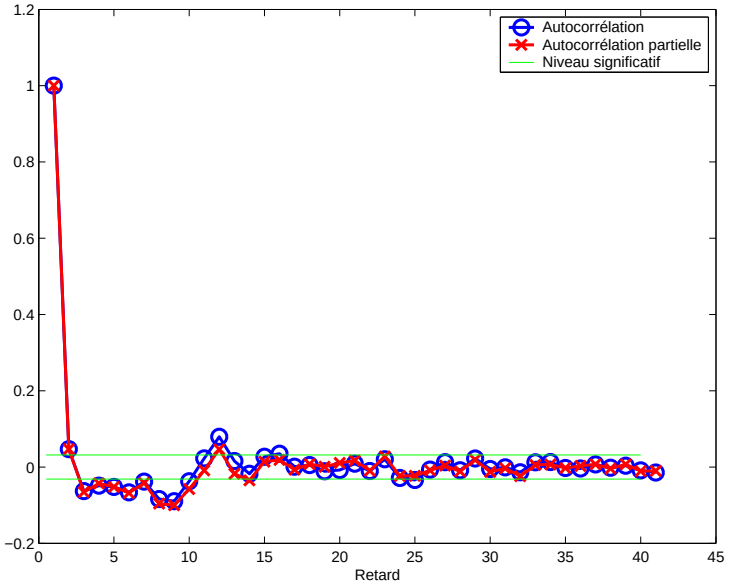
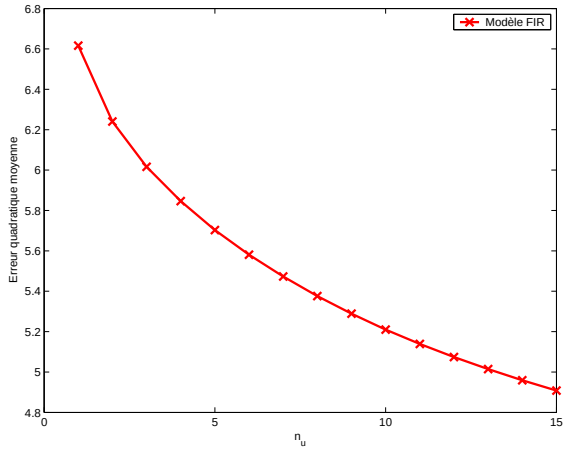


FIGURE 3.5 : Test de linéarité à partir de l'autocorrélation et de l'autocorrélation partielle de l'erreur du modèle OE sur les données « Transistor » sans bruit.

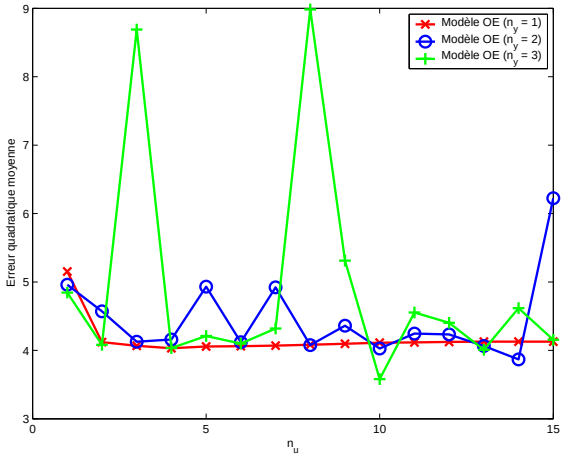
sont à la limite de la stabilité. En se fondant sur les courbes de la figure 3.6(b), le meilleur modèle OE correspond aux ordres $n_u = 10$ et $n_{\hat{y}} = 3$. Toutefois, vu l'irrégularité de la courbe correspondant à $n_{\hat{y}} = 3$, nous avons opté pour un modèle OE légèrement moins performant mais correspondant à une courbe de validation plus régulière, à savoir $n_{\hat{y}} = 1$. Par conséquent, le modèle OE optimal pour les données « Maquette » correspond aux ordres $n_u = 4$ et $n_{\hat{y}} = 1$. Un récapitulatif des performances des différents modèles est donné dans le tableau 3.5. Précisons que pour le modèle ARX, nous avons évalué cette modélisation pour les ordres optimaux du modèle OE à des fins de comparaison, les ordres optimaux pour le modèle ARX étant délicats à déterminer à cause des problèmes d'instabilité rencontrés. L'utilisation de la contrainte de stabilité présente dans l'algorithme d'identification utilisé n'a pas permis de résoudre ce problème de choix des paramètres optimaux. La réponse temporelle des modèles FIR, OE et ARX en comparaison avec les mesures est représentée sur la figure 3.6(c). On constate que les modèles OE et FIR fournissent les meilleurs résultats mais peinent à approximer correctement les régimes permanents des mesures de température. Au final, le meilleur modèle linéaire pour les données « Maquette » est donc un modèle OE avec $n_u = 4$ et $n_{\hat{y}} = 1$. Sur ce modèle, l'autocorrélation de l'erreur obtenue (figure 3.7) confirme les difficultés rencontrées lors de l'apprentissage. Toutefois, l'analyse de l'autocorrelation partielle semble indiquée que le modèle a été correctement identifié.

Type de Modèle	Ordres optimaux	Erreur de validation	Erreur de test
FIR	$n_u = 15$	4.908	3.866
ARX	$n_u = 4$ et $n_y = 1$	-	3.955
OE	$n_u = 4$ et $n_y = 1$	4.032	3.403

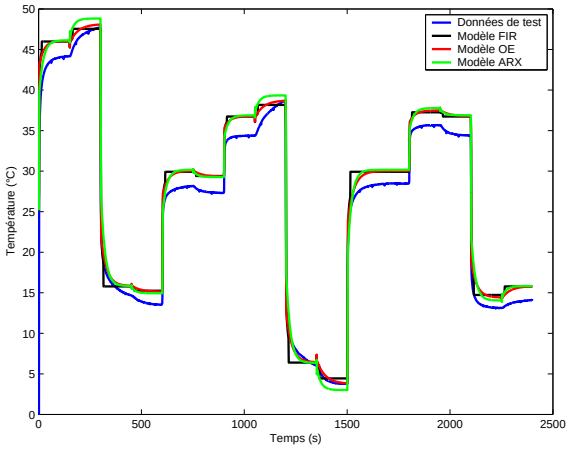
TABLE 3.5 : Récapitulatif des résultats des meilleurs modèles linéaires sur la base de données « Maquette ».



(a) Courbe de validation pour un modèle FIR linéaire.



(b) Courbes de validation correspondant à un modèle OE linéaire.



(c) Comparaison des sorties des modèles linéaires optimaux avec les données.

FIGURE 3.6 : Récapitulatif de la procédure de sélection des modèles linéaires optimaux et performances en test de ces modèles sur la base de données « Maquette ». Notons que les résultats présentés sont obtenus en prenant le même ordre n_u sur les deux entrées (Puissance injectée et Vitesse de refroidissement) de la base.

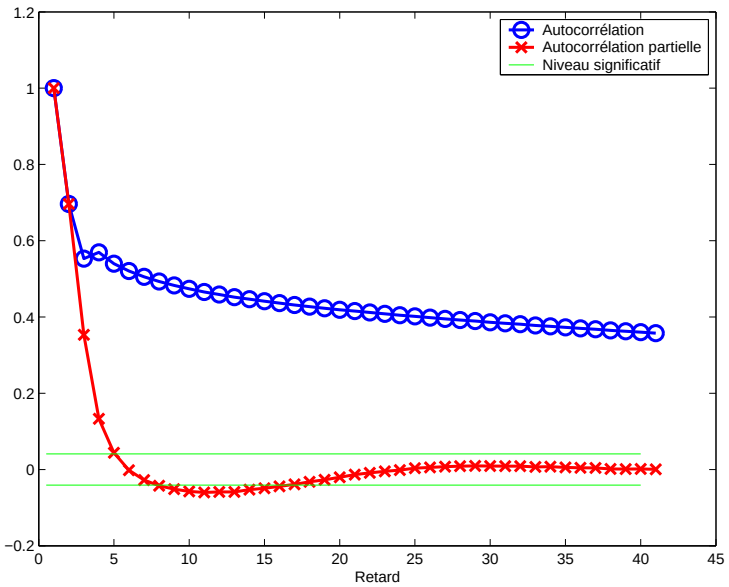


FIGURE 3.7 : Test de linéarité à partir de l'autocorrélation et de l'autocorrélation partielle de l'erreur du modèle OE sur les données « Maquette ».

3.5 Modélisation par réseaux de neurones

3.5.1 Principes

3.5.1.1 Une origine biologique

La désignation de « neurone » provient d'une analogie biologique. Chaque neurone biologique possède la capacité de générer un potentiel d'action au sein de son corps cellulaire. Ce potentiel électrique dans le neurone émetteur est lié à un changement de la concentration en différents ions (K^+ et Na^-) à l'intérieur de la cellule et se propage le long de l'axone jusqu'aux synapses. Le potentiel pré-synaptique génère l'émission de neuromédiateurs vers les récepteurs présents sur la dendrite du neurone récepteur et va modifier le potentiel post-synaptique (à la hausse ou à la baisse selon que la synapse soit inhibitrice ou excitatrice). L'ensemble des signaux ainsi transmis à un neurone vont ensuite être intégrés spatialement et temporellement à l'intérieur du corps cellulaire. Si le potentiel résultant dépasse un certain seuil, le neurone va alors à son tour générer un potentiel d'action.

3.5.1.2 Le neurone artificiel

Lors des débuts vers ce qui allait devenir l'intelligence artificielle, l'une des premières idées a été de copier ce qui fonctionnait déjà, à savoir le cerveau humain et donc à plus petite échelle le neurone. Le fonctionnement du neurone a donc été formalisé de manière à pouvoir le reproduire artificiellement. De cette démarche est né le neurone formel [MP43], qui est donc constitué des éléments suivants :

- Les entrées
  Notées $\mathbf{x}_{i,j}$, elles sont chacune porteuses d'un poids, lui même noté w_j. Ce sont ces poids qui serviront de paramètres au neurone. Pour la suite, on adopte la notation $\mathbf{w}^\top = [w_1, ..., w_D]$ où D est le nombre de poids du neurone.
- La fonction d'agrégation ou potentiel
  Cette fonction $v(\mathbf{w}, \mathbf{x}_i)$, souvent abrégée $v(\mathbf{x}_i)$, combine les différentes entrées et le poids du neurone afin d'obtenir un scalaire, nommé potentiel. Afin de se rapprocher du fonctionnement biologique, cette fonction prend dans la majorité des cas la forme d'une somme pondérée :

  $$v(\mathbf{x}_i) = \sum_{j=1}^{D} w_j \mathbf{x}_{i,j} + b$$

  où $\mathbf{x}_{i,j}$ représente la $j^{\text{ème}}$ composante du vecteur d'entrée $\mathbf{x}_i$. Toutefois, la fonction d'agrégation peut également être représentée par une fonction distance le plus souvent dérivée d'une norme :

  $$v(\mathbf{x}_i) = \|\mathbf{w} - \mathbf{x}_i\| \tag{3.26}$$

- La fonction d'activation
  Cette fonction $a(v)$ va déterminer la sortie s du neurone. Elle peut prendre différentes formes en fonction des besoins du problème. Dans le cas classique du neurone de McCulloch-Pitts [MP43], il s'agit d'une fonction Heaviside :

  $$a(v) = \begin{cases} 0 \text{ si } v \leq 0 \\ 1 \text{ si } v > 0 \end{cases} \tag{3.27}$$

  Cette fonction a par la suite été remplacée par des fonctions dérivables de formes proches telles que la tangente hyperbolique $a(v) = \tanh(v)$ ou la fonction sigmoïde $a(v) = 1/(1+e^{-\alpha v})$. Dans le cas d'une fonction d'activation de type distance, elle prend la forme d'une fonction radiale telle qu'une gaussienne $a(v) = e^{-v^2}$.

Un neurone réalise donc une fonction non linéaire bornée [DMS+02] $s_i = a(v(\mathbf{x}_i, \mathbf{w}))$ où $\mathbf{x}_i$ représente l'entrée du neurone et $\mathbf{w}$ est le vecteur des paramètres (figure 3.8).

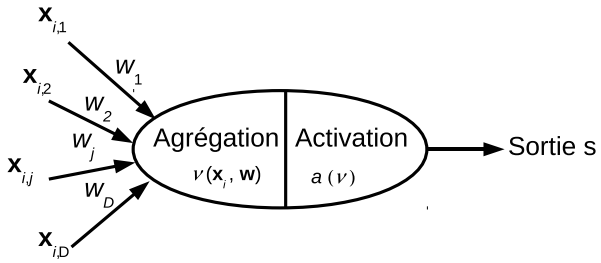


FIGURE 3.8 : Schéma d'un neurone formel

3.5.1.3 Les réseaux de neurones à une couche cachée

Bien qu'intéressant, ce formalisme n'aurait pas connu le succès sans l'apparition des premières règles d'apprentissage pouvant lui être associées. La première d'entre elles est due à Hebb [Heb49] et s'appuie sur un apprentissage par le renforcement du lien entre 2 neurones dont les excitations sont synchrones. Les règles d'adaptation suivantes se rapprochent plus du cadre classique de l'apprentissage que nous avons vu au début de ce chapitre. Pour les illustrer, nous allons considérer le cas d'un seul neurone auquel est soumis un problème de classification défini par une base d'exemples $\{(\mathbf{x}_i, y_i)\}, i \in [1,N]$ où $y_i \in \{+1,-1\}$. La fonction d'agrégation est une somme pondérée.

La sortie du neurone pour le i-ème vecteur d'entrées $\mathbf{x}_i$ est donnée par :

$$s_i = a(\mathbf{w}^\top \mathbf{x}_i + b)$$

On suppose que les poids du neurones sont initialisés aléatoirement, l'idée est désormais de trouver la direction vers laquelle il faut faire tendre ces poids pour diminuer le risque empirique. On rappelle que le rsique empririque à minimiser est :

$$R_e(\mathbf{w}) = \frac{1}{N}\sum_{i=1}^{N} \ell(f(\mathbf{x}_i), y_i)$$

Pour les réseaux de neurones, l'apprentissage s'appuie sur des algorithmes de descente de gradient stochastique [Bot04]. Pour un modèle $f(\mathbf{x})$ caractérisé par le vecteur de paramètres $\mathbf{w}$, à chaque couple $(\mathbf{x}_i, y_i)$, la règle d'adaptation des paramètres par le gradient stochastique est :

$$\mathbf{w} = \mathbf{w} - \eta \nabla_{\mathbf{w}} \ell(f(\mathbf{x}_i), y_i) \tag{3.28}$$

où η représente le taux d'apprentissage de l'algorithme.

L'algorithme du perceptron [Ros58] repose sur ce principe d'adaptation stochastique des paramètres du neurone. La fonction d'activation est supposée être une fonction identité, ce qui donne donc un modèle linéaire $f(\mathbf{x}) = s(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b$. La règle de modification des poids utilisée pour le perceptron s'obtient en considérant un coût charnière particulier, adapté au problème de classification considéré et défini par :

$$\ell(f(\mathbf{x}), y) = \max(0, -y f(\mathbf{x}))$$

Afin de simplifier la notation, posons $\mathbf{w}' = \begin{bmatrix}\mathbf{w}^\top & b\end{bmatrix}^\top$ et $\mathbf{x}'_i = \begin{bmatrix}\mathbf{x}_i^\top & 1\end{bmatrix}^\top$. On en déduit cette expression du gradient

$$\nabla_{\mathbf{w}'} \ell(f(\mathbf{x}), y) = -\begin{cases} 0 & \text{si} \quad y_i f(\mathbf{x}_i) \geq 0 \\ y_i \mathbf{x}'_i & \text{si} \quad y_i f(\mathbf{x}_i) < 0 \end{cases}$$

En combinant cette expression du gradient avec l'équation (3.28), on obtient pour $\eta = 1$ la forme bien connue de l'algorithme du perceptron

$$\mathbf{w}' = \mathbf{w}' + \begin{cases} 0 & \text{si} \quad y_i = \text{sign}(f(\mathbf{x}_i)) \\ y_i \mathbf{x}'_i & \text{si} \quad y_i \neq \text{sign}(f(\mathbf{x}_i)) \end{cases}$$

après échantillonnage de l'observation $(\mathbf{x}_i, y_i)$. Il est établi que l'algorithme du perceptron converge vers une solution optimale en un nombre fini d'itérations lorsque les 2 classes sont linéairement séparables.

Dans le même ordre d'idée, Widrow et Hoff [WH60] proposent une autre règle d'adaptation pour un neurone dont la fonction d'activation est l'identité. Pour le coût ℓ, on utilise une fonction quadratique $\ell(f(\mathbf{x}), y) = (y - f(\mathbf{x}))^2$. On utilise la notation précédente et on écrit $f(\mathbf{x}) = \langle \mathbf{w}', \mathbf{x}' \rangle$. Pour un couple $(\mathbf{x}_i, y_i)$, le gradient du coût par rapport aux paramètres $\mathbf{w}'$ s'écrit alors :

$$\nabla_{\mathbf{w}'} \ell(f(\mathbf{x}_i), y_i) = -2(y_i - f(\mathbf{x}_i))\mathbf{x}'_i$$

La règle d'adaptation de $\mathbf{w}'$ se déduit facilement à partir de (3.28). Cette dernière technique s'adapte particulièrement aux problèmes de régression, ce qui la rapproche des approches d'estimation par moindres carrés récursifs (voir par exemple le livre de Gustafson [Gus00] pour une présentation).

3.5.1.4 Les réseaux de neurones à plusieurs couches cachées

Les algorithmes précédents sont extensibles à des réseaux de neurones multi-couches (figure 3.9). Néanmoins, au début, ces algorithmes ne permettaient d'adapter que la dernière couche du réseau et les commentaires de Minsky et Papert [MP69], notamment sur l'incapacité du perceptron à résoudre le fameux problème du XOR (Ou exclusif), ont marqué un coup d'arrêt important à l'intérêt pour l'approche connexionniste dans l'apprentissage artificiel. Il fallut cependant attendre le début des années 80 pour voir émerger des algorithmes pour les réseaux multi-couches. En effet, même si l'algorithme de rétro-propagation du gradient avait été découvert dès 1974 par Werbos [Wer74], les articles de LeCun [LeC85] et Rumelhart [RHW86] permirent que ce nouvel algorithme suscite l'intérêt qu'il méritait. Même s'il ne permet pas d'obtenir la solution optimale pour les différents poids du réseau, le fait qu'il fonctionne sur des réseaux multi-couches permet d'utiliser les méthodes connexionnistes sur des problèmes où les exemples ne sont pas linéairement séparables.

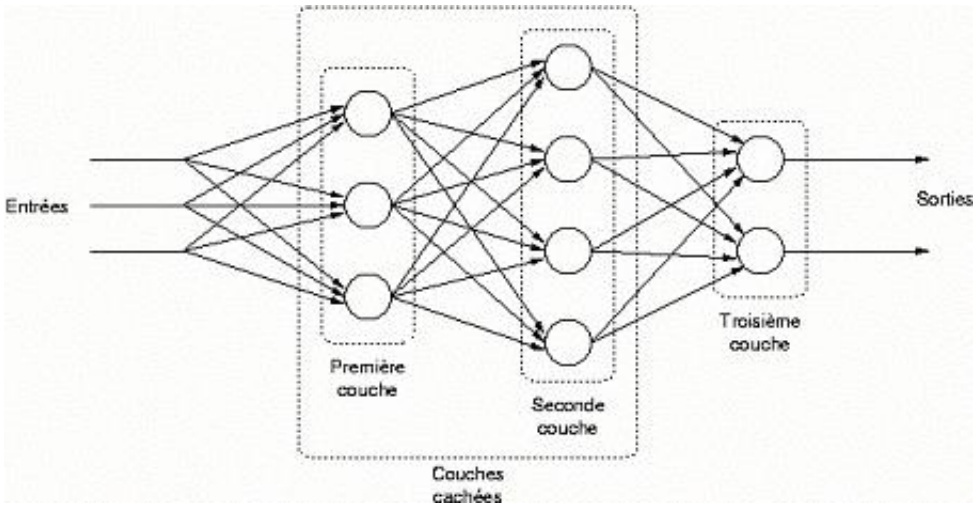


FIGURE 3.9 : Réseau de neurones à 2 couches cachées.

Pour illustrer cet algorithme, nous allons considérer un réseau de neurones avec une seule couche cachée et une seule sortie. Tous les neurones de la couche cachée ont une structure identique, leur fonction d'agrégation est la somme pondérée et leur fonction d'activation est la tangente hyperbolique et seuls leurs poids diffèrent. La couche de sortie est constituée par un neurone linéaire. Les notations afférentes sont définies dans la table 3.6.

De même que pour un seul neurone, on cherche à réduire le risque empirique par adaptation des paramètres du réseau :

$$R_e = \frac{1}{N} \sum_{i=1}^{N} \ell(f(\mathbf{x}_i), y_i)$$

$\mathbf{w}_j^{(l)}$	vecteur des poids entre les neurones de la couche $l-1$ et le j-ème neurone de la couche l
$\mathbf{W}^{(l)}$	matrice des poids entre les neurones de la couche $l-1$ et les neurones de la couche l
$\mathbf{s}^{(l)}$	vecteur des sorties des neurones de la couche l
$\mathbf{b}^{(l)}$	vecteur des biais de la couche l

TABLE 3.6 : Notations utilisées pour les réseaux de neurones multi-couches

De part la définition du réseau, on a :

$$\begin{aligned}\mathbf{s}^{(2)} &= \mathbf{W}^{(2)}\mathbf{s}^{(1)} + \mathbf{b}^{(2)}\\ \mathbf{s}^{(1)} &= \tanh(\mathbf{W}^{(1)}\mathbf{s}^{(0)} + \mathbf{b}^{(1)})\end{aligned}$$

avec :

$$\begin{aligned}\mathbf{s}^{(0)} &= \mathbf{x}\\ \mathbf{s}^{(2)} &= f(\mathbf{x})\end{aligned}$$

Pour la suite $\mathbf{s}_i^{(j)}$ représente l'évaluation du vecteur des sorties des neurones de la couche j pour l'exemple $\mathbf{x}_i$. Pour chaque observation $(\mathbf{x}, y)$, on s'intéresse aux modifications à apporter aux poids $\mathbf{W}^{(1)}$ et $\mathbf{W}^{(2)}$ et aux biais $\mathbf{b}^{(1)}$ et $\mathbf{b}^{(2)}$ afin de réduire le risque lié à l'observation courante. On considère un coût quadratique, étant entendu que la démarche que nous explicitons se généralise facilement à n'importe quelle fonction de coût continue et dérivable.

Dans un premier temps, on cherche les modifications à apporter aux poids $\mathbf{W}^{(2)}, \mathbf{b}^{(2)}$ de la couche de sortie du réseau. Remarquons que dans le contexte de notre exemple (une couche cachée et une couche de sortie à un neurone), on a $\mathbf{W}^{(2)} = \left(\mathbf{w}_1^{(2)}\right)^\top$ et $\mathbf{b}^{(2)} = b^{(2)}$. Le gradient du coût par rapport $\left(\mathbf{w}_1^{(2)}\right)^\top$ est alors :

$$\nabla_{\mathbf{w}_1^{(2)\top}} \ell\left(f(\mathbf{x}_i), y_i\right) = \nabla_{\mathbf{w}_1^{(2)\top}} \ell\left(\mathbf{s}_i^{(2)}, y_i\right).$$

On décompose le gradient du coût associé au couple $(\mathbf{x}_i, y_i)$. Pour ce faire, nous nous appuyons sur des éléments de calcul tensoriel et de dérivation en chaîne exposés dans l'annexe C.

$$\nabla_{\mathbf{w}_1^{(2)\top}} \ell\left(\mathbf{s}_i^{(2)}, y_i\right) = \langle \nabla_{\mathbf{s}_i^{(2)}} \ell\left(\mathbf{s}_i^{(2)}, y_i\right) | \nabla_{\mathbf{w}_1^{(2)\top}} \mathbf{s}_i^{(2)} \rangle$$

où le symbole $\langle \cdot | \cdot \rangle$ représente le produit de dualité (voir section C.4.2 de l'annexe C).

Pour le premier gradient de l'équation précédente, compte tenu de l'utilisation d'un critère quadratique, on déduit :

$$\nabla_{\mathbf{s}_i^{(2)}} \ell\left(\mathbf{s}_i^{(2)}, y_i\right) = 2\left(\mathbf{s}_i^{(2)} - y_i\right). \tag{3.29}$$

Pour le second gradient, on obtient :

$$\nabla_{\mathbf{w}_1^{(2)\top}} \mathbf{s}_i^{(2)} = \frac{\partial \mathbf{s}_i^{(2)}}{\partial \mathbf{w}_1^{(2)\top}} = \mathbf{s}_i^{(1)}.$$

En suivant le même raisonnement, pour $b^{(2)}$, on obtient :

$$\nabla_{b^{(2)}} \ell\left(f(\mathbf{x}_i), y_i\right) = 2\left(\mathbf{s}_i^{(2)} - y_i\right)$$

Pour les poids des neurones de la couche cachée $\mathbf{W}^{(1)}$, si l'on considère le vecteur de sortie $\mathbf{s}_i^{(1)}$ de la couche cachée, on peut écrire :

$$\nabla_{\mathbf{W}^{(1)}} \ell\left(f(\mathbf{x}_i), y_i\right) = \langle \nabla_{\mathbf{s}_i^{(1)}} \ell\left(\mathbf{s}_i^{(2)}, y_i\right) | \nabla_{\mathbf{W}^{(1)}} \mathbf{s}_i^{(1)} \rangle.$$

Dans l'expression de droite de la dernière équation, le premier argument du produit de dualité est assez simple à calculer :

$$\nabla_{\mathbf{s}_i^{(1)}} \ell\left(\mathbf{s}_i^{(2)}, y_i\right) = \langle \nabla_{\mathbf{s}_i^{(2)}} \ell\left(\mathbf{s}_i^{(2)}, y_i\right) | \nabla_{\mathbf{s}_i^{(1)}} \mathbf{s}_i^{(2)} \rangle. \tag{3.30}$$

Or, on connaît déjà $\nabla_{\mathbf{s}_i^{(2)}} \ell\left(\mathbf{s}_i^{(2)}, y_i\right)$ (donné par l'équation 3.29). De plus, on a $\nabla_{\mathbf{s}_i^{(1)}} \mathbf{s}_i^{(2)} = \mathbf{w}_1^{(2)\top}$. On obtient ainsi :

$$\nabla_{\mathbf{s}_i^{(1)}} \ell\left(\mathbf{s}_i^{(2)}, y_i\right) = 2\left(\mathbf{s}_i^{(2)} - y_i\right) \mathbf{w}_1^{(2)\top}.$$

Pour le deuxième argument, on sait que : $\mathbf{s}^{(1)} = \tanh(\mathbf{v}^{(1)})$ avec $\mathbf{v}^{(1)} = \mathbf{W}^{(1)}\mathbf{x} + \mathbf{b}^{(1)}$. On notera ici que la fonction tanh est appliquée à un vecteur $\mathbf{v}$ et est donc prise élément par élément. En utilisant les règles de dérivation chaînée, on obtient :

$$\nabla_{\mathbf{W}^{(1)}} \mathbf{s}^{(1)} = \langle \nabla_{\mathbf{v}^{(1)}} \mathbf{s}^{(1)} | \nabla_{\mathbf{W}^{(1)}} \mathbf{v}^{(1)} \rangle$$

avec

$$\nabla_{\mathbf{v}^{(1)}} \mathbf{s}^{(1)} = diag\left(1 - \tanh^2\left(\mathbf{v}_j^{(1)}\right)\right).$$

On peut remarquer que $\nabla_{\mathbf{v}^{(1)}} \mathbf{s}^{(1)}$ est le gradient d'un vecteur par rapport à un vecteur ce qui donne une matrice. Comme les éléments $\mathbf{v}_j^{(1)}$ de $\mathbf{v}^{(1)}$ sont indépendants des éléments $\mathbf{s}_m^{(1)}$ pour tout $j \neq m$, la matrice jacobienne résultante est diagonale.

Finalement, le dernier terme de la dérivation est obtenue par

$$\nabla_{\mathbf{W}^{(1)}} \mathbf{v}^{(1)} = \left[\frac{\partial \mathbf{v}_j^{(1)}}{\partial \mathbf{W}_{k,m}^{(1)}}\right].$$

L'équation précédente exprime le gradient d'un vecteur par rapport à une matrice et produit donc un tenseur d'ordre 3 dont les éléments sont donnés par le gradient de chaque composante $\mathbf{v}_j^{(1)}$ par rapport aux composantes $\mathbf{W}_{k,m}^{(1)}$ de la matrice de poids de la couche cachée. Notons que des calculs similaires peuvent être aisément conduits pour le vecteur de biais $\mathbf{b}^{(1)}$ et ne seront donc pas détaillés.

L'algorithme se généralise facilement à un nombre de couches quelconque. En effet, considérons un réseau à q couches cachées. Tant que la fonction coût est dérivable, il est possible de calculer pour la couche de sortie $\nabla_{\mathbf{s}_i^{(q)}} \ell\left(\mathbf{s}_i^{(q)}, y_i\right)$. Si l'on généralise l'équation 3.30, on peut écrire :

$$\nabla_{\mathbf{s}_i^{(j)}} \ell\left(\mathbf{s}_i^{(q)}, y_i\right) = \langle \nabla_{\mathbf{s}_i^{(j+1)}} \ell\left(\mathbf{s}_i^{(q)}, y_i\right) | \nabla_{\mathbf{s}_i^{(j)}} \mathbf{s}_i^{(j+1)} \rangle.$$

Or, quelque soit la couche j considérée pour $j \in [1, q]$, on a toujours :

$$\nabla_{\mathbf{W}^{(j)}} \ell\left(\mathbf{s}_i^{(q)}, y_i\right) = \langle \nabla_{\mathbf{s}_i^{(j)}} \ell\left(\mathbf{s}_i^{(q)}, y_i\right) | \nabla_{\mathbf{W}^{(j)}} \mathbf{s}_i^{(j)} \rangle.$$

Si l'on calcule récursivement les $\nabla_{\mathbf{s}_i^{(j)}} \ell\left(\mathbf{s}_i^{(q)}, y_i\right)$ en partant de la couche de sortie, il est alors possible d'adapter tous les poids du réseau. L'information se transmet donc de la couche de sortie vers les couches précédentes. C'est ce sens de propagation de l'information de gradient qui justifie le nom de « rétro-propagation » donné aux algorithmes d'apprentissage des réseaux de neurones multi-couches.

3.5.1.5 Applications des réseaux de neurones aux systèmes dynamiques

La première étape pour réaliser un modèle dynamique à partir d'un réseau de neurones consiste à introduire la dimension temporelle. La méthode la plus simple est de prendre comme entrée du réseau de neurones, le vecteur

de régression φ_k de la même façon que pour les modèles linéaires. Selon le type de vecteur de régression choisi, on obtient une extension non-linéaire naturelle des modèles linéaires (voir tableau 3.3). Ainsi lorsque seules les entrées de commande sont concernées, par analogie avec le cas linéaire, on parle de modèle NNFIR. Les retards peuvent également être introduits sur la sortie réelle et l'on obtient un modèle NNARX. Ces deux modèles ne nécessitent pas d'algorithme particulier car toutes les données nécessaires sont disponibles au moment de l'apprentissage. Ce n'est pas le cas pour les modèles NNARMAX et NNOE (un exemple de ce type de réseau est donnée dans la figure 3.10) où la sortie simulée est également intégrée au vecteur de régression. La rétro-propagation classique ne peut donc plus être utilisée sur ces réseaux récurrents.

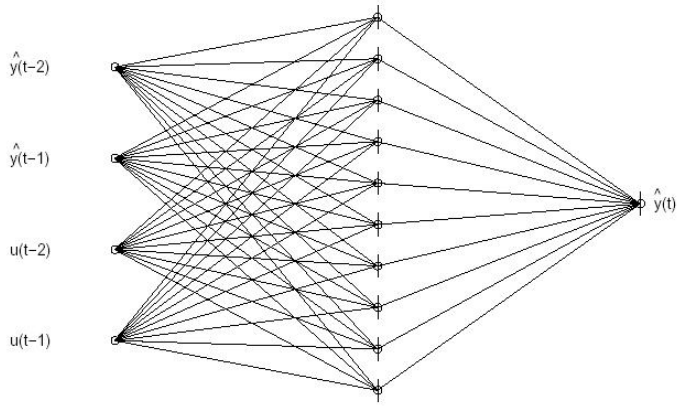


FIGURE 3.10 : Réseau de neurones de type NNOE. Les sorties estimées sont réinjectées en entrée du réseau (ce sont les termes $\hat{y}(t-1)$ et $\hat{y}(t-2)$).

Pour illustrer les algorithmes adaptés aux modèles OE, on choisit l'exemple d'un réseau avec une seule sortie et dont l'ensemble des paramètres sont contenus dans la matrice $\mathbf{W}$. La première alternative consiste à déplier temporellement le réseau. En pratique, ceci revient à dupliquer le réseau « devant » le réseau originel pour obtenir un réseau non-bouclé. La profondeur de ce déploiement doit correspondre à la taille de la base d'apprentissage. Une fois le réseau déplié, on applique l'algorithme de rétro-propagation du gradient. Cette procédure correspond à l'algorithme de « back propagation through time » (BPTT [Wer90]).

Dans ce cas, on a analogie entre l'indice de la couche et l'indice temporel des données $(\mathbf{x}_i, y_i)$. On considère $\mathbf{s}^{(i)}$ le vecteur d'entrée de la couche $i+1$ du réseau. Certains éléments de $\mathbf{s}^{(i)}$ correspondent aux entrées $\mathbf{x}_i$. De plus, le dernier élément correspond à la i-ème réponse du réseau, on peut donc comparer la sortie obtenue $\hat{y}_i$ avec la sortie attendue y_i. Pour les éléments de $\mathbf{s}^{(i)}$ correspondant aux entrées, l'erreur associée est toujours considérée comme nulle. On cherche à réduire le risque empirique que l'on définit par :

$$R_e = \frac{1}{N}\sum_{i=1}^{N} \ell\left(\hat{y}_i, y_i\right)$$

Puisque l'on a choisi de dupliquer N fois le réseau, chaque couche possède sa propre matrice de paramètres $\mathbf{W}^{(i)}$. On cumule alors les gradients obtenus par la formule :

$$\nabla_{\mathbf{W}} R_e = \frac{1}{N}\sum_{i=1}^{N} \nabla_{\mathbf{W}^{(i)}} R_e$$

La règle de dérivation en chaîne donne :

$$\nabla_{\mathbf{W}^{(i)}} R_e = \langle \nabla_{\mathbf{s}^{(i)}} R_e | \nabla_{\mathbf{W}^{(i)}} \mathbf{s}^{(i)} \rangle$$

Dans ce cas, le risque empirique dépend de $\mathbf{s}^{(i)}$ directement par la sortie prédite $\hat{y}_i$ présente dans ce vecteur

mais aussi par son influence sur la sortie de la couche suivante du réseau. On obtient alors :

$$\nabla_{\mathbf{s}^{(i)}} R_e = \frac{\partial R_e}{\partial \mathbf{s}^{(i)}} + \langle \nabla_{\mathbf{s}^{(i+1)}} R_e | \nabla_{\mathbf{s}^{(i)}} \mathbf{s}^{(i+1)} \rangle.$$

Par le même processus de calcul récursif que dans la rétropropagation classique, il est alors possible de calculer la modification à apporter aux poids $\mathbf{W}$ pour diminuer le risque empirique. Toutefois, pour conserver des réseaux de taille raisonnable, on peut se contenter en pratique d'une profondeur de déploiement d'ordre n plus faible que la taille de la base N. Cette restriction est d'ailleurs justifiée par le phénomène d'« évanouissement du gradient » dans les réseaux possédant trop de couches cachées [BSF94].

Le deuxième algorithme pour les modèles NNOE ne considère pas de duplication du réseaux. La dérivation directe de R_e par rapport à $\mathbf{W}$ donne :

$$\nabla_{\mathbf{W}} R_e = \frac{1}{N} \sum_{i=1}^{N} \nabla_{\mathbf{W}} \ell(\hat{y}_i, y_i).$$

En utilisant une nouvelle fois la règle de dérivation en chaîne, on obtient :

$$\nabla_{\mathbf{W}} \ell(\hat{y}_i, y_i) = \langle \nabla_{\mathbf{s}^{(i)}} \ell(\hat{y}_i, y_i) | \nabla_{\mathbf{W}} \mathbf{s}^{(i)} \rangle.$$

On calcule le premier élément du second membre de l'équation par la formule :

$$\nabla_{\mathbf{s}^{(i)}} \ell(\hat{y}_i, y_i) = \frac{\partial \ell(\hat{y}_i, y_i)}{\partial \mathbf{s}^{(i)}}$$

Pour le second élément, on utilise également la règle de dérivation en chaîne pour écrire :

$$\nabla_{\mathbf{W}} \mathbf{s}^{(i)} = \frac{\partial \mathbf{s}^{(i)}}{\partial \mathbf{W}} + \langle \nabla_{\mathbf{s}^{(i-1)}} \mathbf{s}^{(i)} | \nabla_{\mathbf{W}} \mathbf{s}^{(i-1)} \rangle.$$

Les gradients nécessaires se calculent récursivement mais la transmission de l'information se fait dans le sens classique de transmission pour les réseaux de neurones. L'initialisation est effectuée en considérant que l'état initial du réseau ne dépend pas des paramètres et donc $\nabla_{\mathbf{W}} \mathbf{s}^0 = [0]$. Cet algorithme, nommé « real time recurrent learning » (RTRL [WZ89]), permet donc lui aussi de calculer récursivement les adaptations à apporter aux poids $\mathbf{W}$ du réseau.

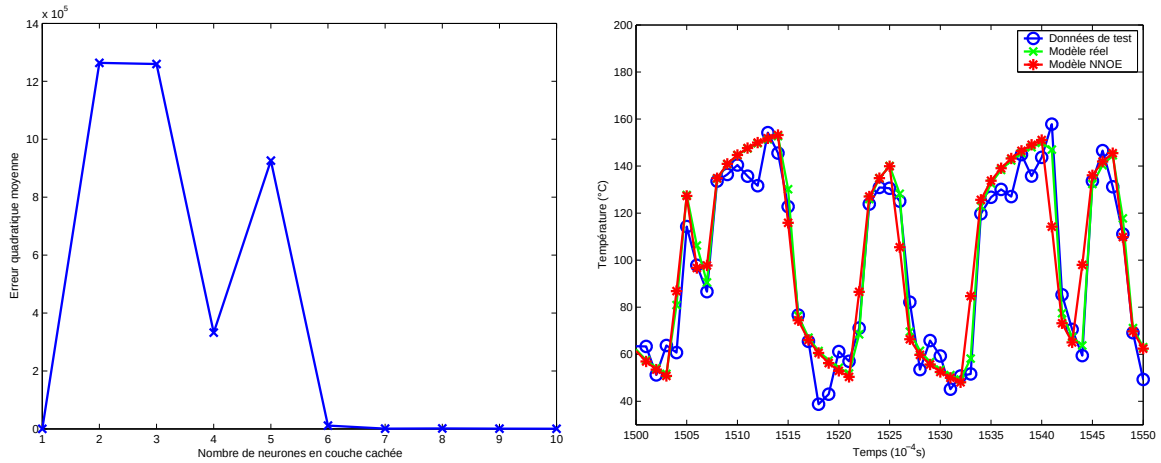
La majorité des algorithmes d'apprentissage de modèles neuronaux OE sont des variantes des deux techniques présentées en essayant d'exploiter et de combiner leurs avantages respectifs.

3.5.2 Prédiction de température par les réseaux de neurones dynamiques

Les réseaux utilisés ont été générés grâce à la toolbox NNSYSID ([NRPH00]) (Neural Network Based System Identification Toolbox) qui utilise une méthode de type RTRL pour apprendre des modèles dynamiques. Il s'agit de perceptrons à 1 couche cachée, la couche de sortie étant linéaire. Ces modèles constituent en effet une classe d'approximateurs universels parcimonieux [HSW89, HSWA], il n'est donc pas utile de rechercher parmi les réseaux possédant plus de couches. On reprend les vecteurs de régression obtenus dans le cas linéaire (1 retard sur la sortie et 4 sur l'entrée pour les données « Transistor » et de même pour les données « Maquette »). Le seul paramètre restant à choisir est le nombre de neurones en couche cachée. On compare donc l'erreur quadratique moyenne obtenue pour les différentes architectures en utilisant la même procédure de validation croisée que dans le cas linéaire.

Pour les données « Transistor », on constate que le meilleur résultat est obtenu pour un seul neurone en couche cachée (figure 3.11(a)). Ce résultat n'est pas étonnant compte tenu des bons résultats déjà obtenus par les modèles

linéaires. Il ne fait que confirmer la faible non-linéarité présente dans le système. La sortie obtenue est donc fortement similaire à celle obtenue pour le modèle linéaire (figure 3.11(b) et la différence maximale entre les 2 modèles ne dépasse pas 0.5°C) même si l'on constate une légère différence de l'erreur quadratique moyenne en apprentissage ($J_{val} = 191.878$) et sur la base de test ($J_{test} = 185.740$ pour les données avec bruit et $J_{test} = 80.152$ pour les données sans bruit) en faveur du modèle NNOE (voir tableau 3.4 pour les performances des modèles linéaires).

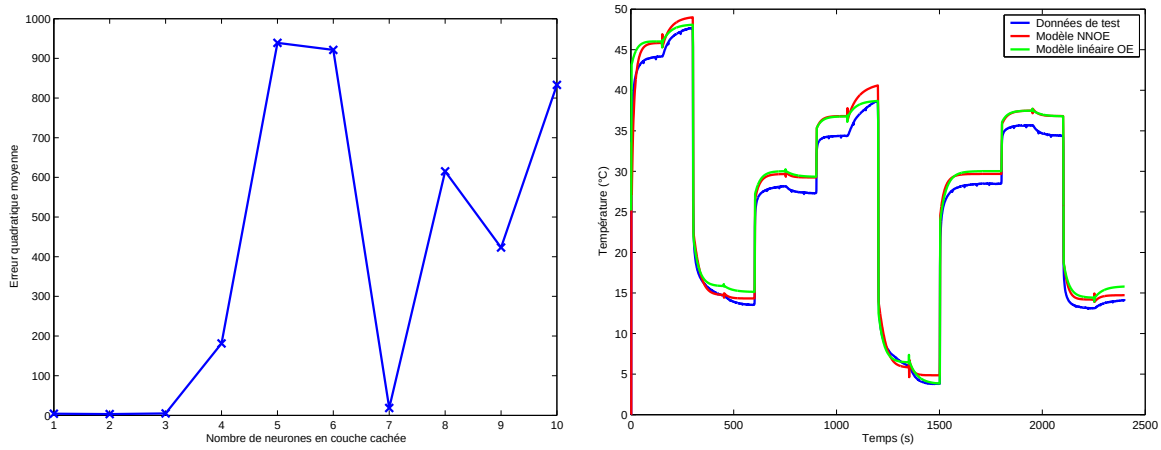


(a) Variation de l'erreur de validation en fonction du nombre de neurones de la couche cachée.

(b) Comparaison de la sortie du modèle NNOE avec les données « Transistor » avec et sans bruit.

FIGURE 3.11 : Résultats obtenus pour un modèle NNOE sur la base de données « Transistor »

Pour les données « Maquette », on constate que le meilleur résultat est obtenu pour un réseau comportant deux neurones en couche cachée, comme le montre la figure 3.12(a). De plus, l'algorithme d'apprentissage a beaucoup de difficultés à converger sur ces données pour les réseaux possédant plus de neurones. Là encore, les résultats sont un peu meilleurs que ceux des modèles linéaires (erreur de validation $J_{val} = 3.229$ et erreur de test $J_{test} = 2.535$) et le tracé de la réponse confirme une meilleure adéquation avec la dynamique du système réel (figure 3.12(b)). L'emploi de réseaux de neurones semble donc pallier partiellement les lacunes constatées sur les modèles linéaires. La difficulté à modéliser ce système pourrait toutefois ne pas être entièrement liée aux non-linéarités mais aux fortes disparités de dynamiques dans la réponse aux deux entrées (ventilateur beaucoup plus lent et d'influence plus faible que la puissance injectée sur la température). L'emploi d'autres méthodes non-linéaires pourra confirmer cette hypothèse.



(a) Variation de l'erreur de validation en fonction du nombre de neurones de la couche cachée.

(b) Comparaison de la sortie du modèle NNOE avec les données réelles et le meilleur modèle OE linéaire

FIGURE 3.12 : Résultats obtenus pour un modèle NNOE sur la base de données « Maquette »

3.6 Modélisation par SVM

3.6.1 Introduction au principe des SVM pour la classification

Les SVM (Support Vector Machines ou Séparateur à Vaste Marge) sont issus des travaux de Vapnik sur l'apprentissage ([Vap98]). Pour illustrer le concept sous-tendant cette approche, nous nous plaçons dans le cas d'un problème de classification binaire. Sans perte de généralités, on considère que l'espace de sortie est $\mathscr{Y} = \{-1, +1\}$. En supposant l'espace des entrées $\mathscr{X}$ muni du produit scalaire et d'une norme induite, on recherche une fonction de décision linéaire $f(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b$ caractérisée par les paramètres $\mathbf{w} \in \mathscr{X}$ et $b \in \mathbb{R}$. Etant donné une observation $\mathbf{x}$, la prédiction de sa classe est obtenue en prenant le signe $\text{sign}(f(\mathbf{x}))$ de $f(\mathbf{x})$.

Soit un ensemble de données d'apprentissage $\{(\mathbf{x}_i, y_i)\}_{i=1,\cdots,N}$. Pour simplifier notre présentation, considérons que le problème de classification est linéairement séparable c'est-à-dire qu'il existe au moins une fonction de décision linéaire séparant parfaitement les deux classes. Le principe des SVM consiste à définir la fonction de décision $f(\mathbf{x})$ telle que $f(\mathbf{x}_i) > 0$ si $y_i = +1$ et $f(\mathbf{x}_i) < 0$ si $y_i = -1$ tout en gardant une séparation maximale entre les deux classes matérialisée par la marge. La marge est alors définie comme la distance minimale, prise perpendiculairement à l'hyperplan $f(\mathbf{x}) = 0$, entre deux exemples appartenant à des classes différentes. Une illustration de cette notion de marge est représentée à la figure 3.13. La distance entre un point $\mathbf{x}$ et l'hyperplan séparateur peut

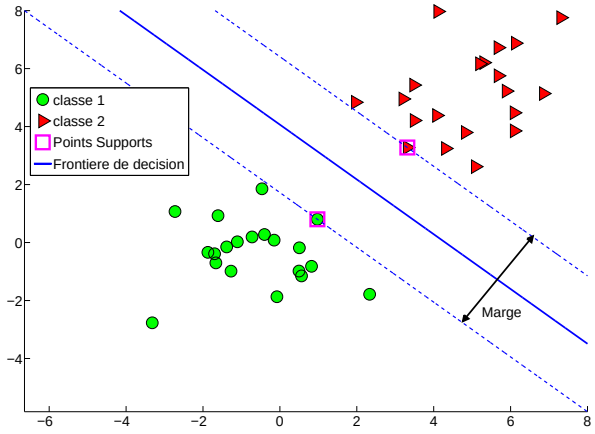


FIGURE 3.13 : Illustration du concept de SVM sur un problème linéairement séparable

alors être écrite comme $d(\mathbf{x}) = \frac{|\mathbf{w}^\top \mathbf{x}|}{\|\mathbf{w}\|}$. Les SVM reposent sur l'utilisation d'un hyperplan canonique c'est-à-dire par normalisation des paramètres $\mathbf{w}$ et b, les points les plus proches de l'hyperplan sont à une distance $d(\mathbf{x}) = \frac{1}{\|\mathbf{w}\|}$. Par conséquent la marge vaut $\frac{2}{\|\mathbf{w}\|}$. Maximiser la marge revient donc à minimiser la norme w. Le problème, dit primal, de classification par un SVM linéaire s'écrit alors [SS01] :

$$\min_{\mathbf{w},b} \quad \frac{1}{2}\|\mathbf{w}\|^2 \tag{3.31a}$$

$$\text{sous les contraintes} \quad y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1, \quad \forall i = 1, \cdots, N \tag{3.31b}$$

En utilisant la méthode des multiplicateurs de Lagrange pour intégrer les contraintes d'inégalité (3.31b), on obtient le Lagrangien :

$$\mathscr{L}(\mathbf{w}, b, \alpha) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{N} \alpha_i [y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) - 1]$$

avec $\alpha_i \geq 0, \forall i = 1, \cdots, N$. La méthode des multiplicateurs de Lagrange recommande de minimiser $\mathscr{L}$ par rapport aux variables primales $(\mathbf{w}, b)$ et maximiser $\mathscr{L}$ par rapport aux multiplicateurs de Lagrange α_i. Les conditions

d'optimalités par rapport aux variables primales donnent alors :

$$\nabla_{\mathbf{w}}\mathscr{L} = 0 \quad \Longrightarrow \quad \mathbf{w} = \sum_{i=1}^{N} \alpha_i y_i \mathbf{x}_i, \tag{3.32}$$

$$\nabla_{b}\mathscr{L} = 0 \quad \Longrightarrow \quad \sum_{i=1}^{N} \alpha_i y_i = 0. \tag{3.33}$$

En intégrant ces dernières équations dans l'expression du lagrangien, on obtient une reformulation du problème qui peut alors être réécrit sous sa forme duale [SS01] :

$$\begin{aligned} \max_{\alpha} \quad & -\frac{1}{2}\sum_{i=1,j=1}^{N} \alpha_i \alpha_j y_i y_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle + \sum_{i=1}^{N} \alpha_i \\ \text{s.c.} \quad & \alpha_i \geq 0, \quad \forall i = 1, \cdots, N \\ & \sum_{i}^{N} \alpha_i y_i = 0 \end{aligned} \tag{3.34}$$

La résolution de ce problème de programmation quadratique permet d'obtenir le vecteur α duquel on déduit $\mathbf{w}$ en utilisant la condition d'optimalité (3.32). Le biais b s'obtient à partir des conditions de Karush-Kuhn-Tucker (KKT). En effet, il est à remarquer qu'à l'optimalité, les conditions KKT supplémentaires [BV04] stipulent que $\alpha_i \left[y_i(\mathbf{w}^\top \mathbf{x}_i + b) - 1 \right] = 0, \quad \forall i = 1, \cdots, N$. Ceci a deux implications :

- les points pour lesquels la contrainte inégalité (3.31b) est strictement satisfaite satisfont $\alpha_i = 0$,
- les points pour lesquels la contrainte (3.31b) est active (c'est-à-dire vérifiant $y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) = 1$) ont leur multiplicateur α_i libre. Seuls ces points participent à la solution (3.32) d'où leur nom de points supports.

Dans certaines références comme [SS01], les conditions KKT relatives aux points supports sont utilisées pour déterminer le biais.

3.6.2 Cas non linéairement séparable

Lorsque les données ne sont plus linéairement séparables, on ne peut plus trouver de classifieur linéaire qui ne commette pas d'erreurs. Si l'on reprend l'idée d'une frontière de décision munie d'une marge, ces erreurs peuvent être de 2 types :

- soit l'exemple est du bon côté de la frontière de décision mais dans la marge,
- soit l'exemple est du mauvais côté de la frontière de décision.

Pour formaliser ces différentes erreurs, on introduit de nouvelles variables positives ξ_i qui vont venir relâcher les contraintes du problème. Ainsi, on remplace les contraintes originelles (3.31b) par :

$$\begin{aligned} y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \quad &\geq \quad 1 - \xi_i, \quad \forall i = 1, \cdots, N \\ \xi_i \quad &\geq \quad 0, \qquad \forall i = 1, \cdots, N \end{aligned}$$

Si $0 \leq \xi_i \leq 1$ alors le point reste bien classé mais est situé dans la marge. Si $\xi_i > 1$ alors le point est mal classé. On va alors chercher à résoudre le problème de classification maximisant la marge et minimisant les erreurs ξ_i. On introduit ici un hyper-paramètre C positif qui va représenter le compromis entre la minimisation des erreurs et la maximisation de la marge. On obtient alors un nouveau problème primal :

$$\begin{aligned} \min_{\mathbf{w},b,\xi} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_i \xi_i \\ \text{s.c.} \quad & y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1 - \xi_i, \qquad \forall i = 1, \cdots, N \\ & \xi_i \leq 0, \qquad \forall i = 1, \cdots, N \end{aligned}$$

En appliquant le principe des multiplicateurs de Lagrange, on établit que l'expression du Lagrangien est :

$$\mathscr{L}(\mathbf{w}, b, \xi, \alpha, \gamma) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_i \xi_i - \sum_{i=1}^{N} \alpha_i \left[y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) - 1 \right] - \sum_{i=1}^{N} \gamma_i \xi_i$$

avec $\alpha_i \geq 0$, $\gamma_i \geq 0$, $\forall i = 1, \cdots, N$ les multiplicateurs de Lagrange. On en déduit les conditions d'optimalité :

$$\begin{aligned}
\nabla_{\mathbf{w}} \mathscr{L} = 0 \quad &\Longrightarrow \quad \mathbf{w} = \sum_{i=1}^{N} \alpha_i y_i \mathbf{x}_i, \\
\nabla_b \mathscr{L} = 0 \quad &\Longrightarrow \quad \sum_{i=1}^{N} \alpha_i y_i = 0, \\
\nabla_{\xi_i} \mathscr{L} = 0 \quad &\Longrightarrow \quad \gamma_i = C - \alpha_i \quad \Longrightarrow 0 \leq \alpha_i \leq C.
\end{aligned}$$

Le problème dual associé peut alors être obtenu de la même manière que précédemment et prend la forme :

$$\begin{aligned}
\max_{\alpha} \quad & -\frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle + \sum_i \alpha_i \\
\text{s.c.} \quad & 0 \leq \alpha_i \leq C, \quad \forall i = 1, \cdots, N \\
& \sum_{i}^{N} \alpha_i y_i = 0
\end{aligned}$$

On remarquera que ce problème dual ne diffère du précédent (3.34) que par l'adjonction d'une borne supérieure C aux paramètres de Lagrange α_i. Hormis ce point, les paramètres $\mathbf{w}$ et b s'obtiennent comme dans le cas linéairement séparable. Une illustration de la solution fournie par un SVM à un problème non-linéairement séparable est donnée à la figure 3.14.

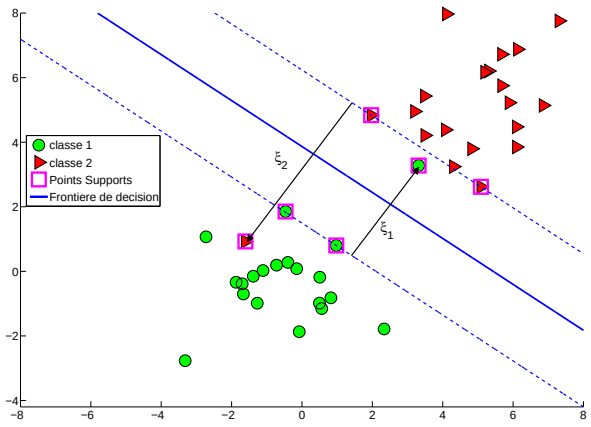


FIGURE 3.14 : Illustration du concept de SVM sur un problème non-linéairement séparable. On remarque qu'en plus des points supports sur la marge, d'autres types de points supports sont à considérer : points mal classés ou à l'intérieur de la marge.

3.6.3 Classification SVM non linéaire

Le relâchement des contraintes permet de traiter les cas non linéairement séparables mais ne permet pas d'obtenir une frontière de décision non-linéaire dans l'espace $\mathscr{X}$ des entrées. L'idée est donc de transformer l'algorithme afin de pouvoir traiter la classification non-linéaire permettant d'obtenir des frontières de décision non-linéaires dans $\mathscr{X}$. Cette modification consiste à projeter les points d'apprentissage de l'espace de départ vers un autre espace, appelé espace des caractéristiques (*feature space*), souvent de dimension plus importante et muni d'un produit scalaire. L'espoir est qu'une fois les points projetés dans ce nouvel espace, les classes puissent être séparées linéairement. La fonction de décision est alors de la forme : $f(\mathbf{x}) = \langle \mathbf{w}, \phi(\mathbf{x}) \rangle + b$, où $\langle ., . \rangle$ désigne le produit scalaire dans l'espace des caractéristiques (*feature space*) et $\phi(\mathbf{x})$ la projection du point $\mathbf{x}$ dans cet espace.

Or, dans sa version linéaire, on voit que l'algorithme des SVM laisse apparaître le produit scalaire $\langle \mathbf{x}_i, \mathbf{x}_j \rangle$. Par conséquent, les algorithmes précédemment exposés s'appliquent aisément au cas non-linéaire en remplaçant

le produit scalaire $\langle \mathbf{x}_i, \mathbf{x}_j \rangle$ par $\langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle$. En choisissant astucieusement le *feature space*, on peut obtenir un produit scalaire relativement simple à calculer directement à partir des données. Il peut alors être réécrit sous la forme : $\langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle = K(\mathbf{x}_i, \mathbf{x}_j)$ où $K(\cdot,\cdot)$ doit être un noyau qui réponde à la condition de Mercer [Mer09], c'est-à-dire une fonction continue, symétrique et semi-définie positive. Cette condition garantit que la fonction $K(\cdot,\cdot)$ définit bien un produit scalaire et induit une norme dans un espace de Hilbert. Ainsi, si la condition de Mercer est vérifiée, la fonction ϕ n'a plus besoin d'être spécifiée. La forme de la frontière de décision dépendra alors du type de noyau choisi. Un aperçu de différentes fonctions noyau communément utilisées est présenté dans le tableau 3.7.

Nom de la fonction noyau	Expression mathématique	Hyper-paramètres
Linéaire	$K(\mathbf{x},\mathbf{z}) = \langle \mathbf{x},\mathbf{z} \rangle$	Aucun
Polynomial	$K(\mathbf{x},\mathbf{z}) = (\langle \mathbf{x},\mathbf{z} \rangle)^n$ ou $(\mathbf{x}^\top \mathbf{z} + c)^n$	n, c
Gaussien	$K(\mathbf{x},\mathbf{z}) = e^{-\frac{\|\mathbf{x}-\mathbf{z}\|^2}{2\sigma^2}}$	σ
Sigmoïde	$K(\mathbf{x},\mathbf{z}) = \tanh(a(\langle \mathbf{x},\mathbf{z} \rangle - b))$	a, b

TABLE 3.7 : Aperçu des différents types de noyaux couramment utilisés. Les points $\mathbf{x}$ et $\mathbf{z}$ appartiennent à l'espace des entrées $\mathscr{X}$. La fonction noyau est définie comme $K(\mathbf{x},\mathbf{z}) : \mathscr{X} \times \mathscr{X} \longrightarrow \mathbb{R}$ qui à toute paire $(\mathbf{x},\mathbf{z})$ associe $K(\mathbf{x},\mathbf{z})$.

3.6.4 Application des SVM à la régression : coût ε-insensible

On s'intéresse maintenant à un problème de régression défini sur la base d'apprentissage $\{(\mathbf{x}_i, y_i)\}_{i=1,\cdots,N}$ avec $y_i \in \mathbb{R}$. Nous nous plaçons dans le cadre non-linéaire où la fonction f est définie de manière générique par $f(\mathbf{x}) = \langle \mathbf{w}, \phi(\mathbf{x}_i) \rangle + b$. Le coût utilisé est un coût ε-insensible (voir figure 3.1). Ce coût dont nous rappelons l'expression ici $\ell(f(\mathbf{x}), y) = \max(0, \varepsilon - |y - f(\mathbf{x})|)$ considère que le risque empirique est nul si l'écart absolu entre y_i et sa prédiction $f(\mathbf{x}_i)$ est inférieure à un certain seuil ε fixé par l'utilisateur. Afin d'obtenir la fonction la plus régulière possible, on cherche toujours à minimiser la norme de $\mathbf{w}$ tout en respectant les contraintes définies par la fonction coût. En imitant le raisonnement conduit pour le problème de classification linéairement séparable, si on veut une fonction f de risque empirique nul, il faut résoudre alors le problème :

$$\begin{aligned} \min_{\mathbf{w},b} \quad & \frac{1}{2}\|\mathbf{w}\|^2 \\ \text{s.c.} \quad & |y_i - (\langle \mathbf{w}, \phi(\mathbf{x}_i) \rangle + b)| \leq \varepsilon, \qquad \forall i = 1, \cdots, N \end{aligned}$$

Tout comme dans le cas de la classification, nous allons relâcher ces contraintes par l'introduction de variables d'écart ξ_i et ξ_i' (correspondant respectivement à une erreur par excès ou par défaut de la prédiction vis-à-vis de la valeur réelle et du seuil ε). On obtient alors le problème primal suivant :

$$\begin{aligned} \min_{\mathbf{w},b,\xi,\xi'} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N}(\xi_i + \xi_i') \\ \text{s.c.} \quad & y_i - (\langle \mathbf{w}, \phi(\mathbf{x}_i) \rangle + b) \leq \varepsilon + \xi_i, \qquad \forall i = 1, \cdots, N \\ & (\langle \mathbf{w}, \phi(\mathbf{x}_i) \rangle + b) - y_i \leq \varepsilon + \xi_i', \qquad \forall i = 1, \cdots, N \\ & \xi_i \geq 0, \ \xi_i' \geq 0, \qquad \forall i = 1, \cdots, N \end{aligned}$$

Le Lagrangien correspondant prend alors la forme :

$$\begin{aligned} \mathscr{L}(\mathbf{w}, b, \xi, \xi', \alpha, \alpha', \gamma, \gamma') \quad = \quad & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N}(\xi_i + \xi_i') - \sum_{i=1}^{N}\alpha_i[\varepsilon + \xi_i - y_i + (\langle \mathbf{w}, \phi(\mathbf{x}_i) \rangle + b)] \\ & - \sum_{i=1}^{N}\alpha_i'[\varepsilon + \xi_i' + y_i - (\langle \mathbf{w}, \phi(\mathbf{x}_i) \rangle + b)] - \sum_{i=1}^{N}\gamma_i\xi_i - \sum_{i=1}^{N}\gamma_i'\xi_i' \end{aligned}$$

avec $\alpha_i \geq 0$, $\alpha'_i \geq 0$, $\gamma_i \geq 0$ et $\gamma'_i \geq 0$, $\forall i = 1, \ldots, N$. Les conditions d'optimalité deviennent alors

$$\begin{aligned}
\nabla_{\mathbf{w}} \mathscr{L} = 0 \quad &\Longrightarrow \quad \mathbf{w} = \sum_{i=1}^{N} (\alpha_i - \alpha'_i)\phi(\mathbf{x}_i), \\
\nabla_b \mathscr{L} = 0 \quad &\Longrightarrow \quad \sum_{i=1}^{N} (\alpha_i - \alpha'_i) = 0, \\
\nabla_{\xi_i} \mathscr{L} = 0 \quad &\Longrightarrow \quad \gamma_i = C - \alpha_i \quad \Longrightarrow \quad 0 \leq \alpha_i \leq C, \\
\nabla_{\xi'_i} \mathscr{L} = 0 \quad &\Longrightarrow \quad \gamma'_i = C - \alpha'_i \quad \Longrightarrow \quad 0 \leq \alpha'_i \leq C.
\end{aligned}$$

En intégrant les conditions d'optimalité dans l'expression du lagrangien, on obtient la formulation duale suivante :

$$\begin{aligned}
\max_{\alpha, \alpha'} \quad & \sum_{i=1}^{N} y_i(\alpha_i - \alpha'_i) - \varepsilon \sum_{i=1}^{N} (\alpha_i + \alpha'_i) - \frac{1}{2} \sum_{i=1, j=1}^{N} (\alpha_i - \alpha'_i)(\alpha_j - \alpha'_j) K(\mathbf{x}_i, \mathbf{x}_j) \\
\text{s.c.} \quad & 0 \leq \alpha_i \leq C, \quad \forall i = 1, \cdots, N \\
& 0 \leq \alpha'_i \leq C, \quad \forall i = 1, \cdots, N \\
& \sum_{i=1}^{N} (\alpha_i - \alpha'_i) = 0
\end{aligned}$$

Une fois les vecteurs α et α' obtenus à partir de la résolution du problème dual, les paramètres $\mathbf{w}$ se déduisent des conditions d'optimalité. Le biais b s'obtient à partir des conditions KKT tout comme dans le cas de la classification. L'expression de la fonction de régression finale est alors

$$\begin{aligned}
f(\mathbf{x}) &= \langle \mathbf{w}, \phi(\mathbf{x}) \rangle + b \\
f(\mathbf{x}) &= \sum_{i=1}^{N} (\alpha_i - \alpha'_i) K(\mathbf{x}_i, \mathbf{x}) + b
\end{aligned}$$

Dans le cas du SVR (Support Vector Regression), tous les points points situés dans la marge du modèle (le « tube » formé par ε autour de la courbe de sortie du modèle) ne sont pas points-supports du modèle. Les coeefi-cients α_i et α'_i qui leur correspond sont nuls. Seul un certain nombre de points, situés sur le tube ou à l'extérieur du tube, sont utilisés pour calculer la sortie du modèle (voir figure 3.15). Cette caractéristique fait du SVR un modèle parcimonieux pour la régression.

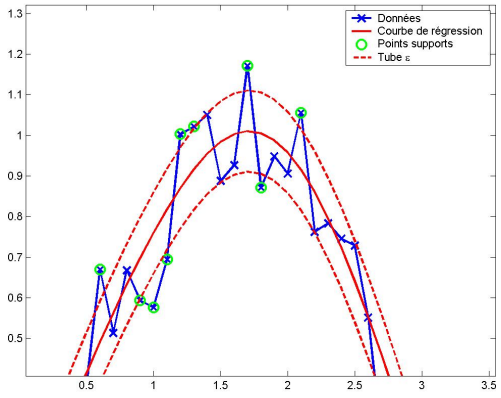


FIGURE 3.15 : Application des SVM à la régression. L'algorithme tend à conserver un maximum de points dans un "tube" créé autour de la fonction estimée par la fonction de coût ε-insensible.

3.6.5 Application des SVM à la régression : coût quadratique (Least Squares SVM)

Une alternative possible pour un problème de régression avec un SVM consiste à remplacer le coût ε-insensible par un coût quadratique (la même idée peut être appliquée à la classification). L'algorithme porte alors le nom de

Least Squares - SVM [SV99] et est dénommé par l'acronyme LS-SVM. On cherche toujours une solution de la forme $f(\mathbf{x}) = \langle \mathbf{w}, \phi(\mathbf{x}) \rangle + b$. Le problème à résoudre prend la forme :

$$\min_{\mathbf{e},\mathbf{w},b} \quad \frac{1}{2}\|\mathbf{w}\|^2 + \frac{C}{2}\sum_i e_i^2 \tag{3.35a}$$

$$\text{s.c.} \quad y_i - (\langle \mathbf{w}, \phi(\mathbf{x}_i) \rangle + b) = e_i \qquad \forall i = 1, \cdots, N \tag{3.35b}$$

On constate que les contraintes du problème sont désormais des contraintes d'égalité ce qui permet de résoudre le problème comme un simple système d'équations linéaires. En effet, le Lagrangien s'écrit :

$$\mathscr{L}(\mathbf{w}, b, e, \alpha) = \frac{1}{2}\|\mathbf{w}\|^2 + \frac{C}{2}\sum_i e_i^2 + \sum_i \alpha_i \left[y_i - (\langle \mathbf{w}, \phi(\mathbf{x}_i) \rangle + b) - e_i \right]$$

En dérivant le Lagrangien par rapport aux variables primales, on obtient :

$$\nabla_{\mathbf{w}} \mathscr{L} = 0 \quad \Rightarrow \quad \mathbf{w} = \sum_i \alpha_i \phi(\mathbf{x}_i) \tag{3.36a}$$

$$\nabla_b \mathscr{L} = 0 \quad \Rightarrow \quad \sum_i \alpha_i = 0 \tag{3.36b}$$

$$\nabla_{e_i} \mathscr{L} = 0 \quad \Rightarrow \quad C e_i - \alpha_i = 0 \tag{3.36c}$$

Si l'on remplace l'expression de $\mathbf{w}$ (3.36a) dans les contraintes égalité (3.35b), on peut écrire ces contraintes sous la forme

$$y_i = \sum_{j=1}^{N} \alpha_j \langle \phi(\mathbf{x}_i, \mathbf{x}_j \rangle + b + e_i = \sum_{j=1}^{N} \alpha_j K(\mathbf{x}_i, \mathbf{x}_j) + b + e_i, \quad \forall i = 1, \cdots, N.$$

On en déduit l'expression matricielle

$$\mathbf{y} = \mathbf{K}\alpha + b\mathbb{1} + \mathbf{e}. \tag{3.37}$$

avec $\mathbf{K} \in \mathbb{R}^{N \times N}$ la matrice de Gram dont les éléments sont les termes $K(\mathbf{x}_i, \mathbf{x}_j)$. Le vecteur $\mathbf{e} \in \mathbb{R}^N$ contient tous les termes de résidus $y_i - (\langle \mathbf{w}\phi(\mathbf{x}_i) \rangle + b)$. Le vecteur $\mathbf{y} \in \mathbb{R}^N$ comprend toutes les sorties y_i alors que $\mathbb{1}$ désigne un vecteur composé de 1 et de dimension appropriée. En compilant les conditions d'optimalité (3.36c), on peut également établir la relation matricielle $C\mathbf{e} = \alpha$ qui permet d'éliminer $\mathbf{e}$ de (3.37). A partir de cette observation et en intégrant la contrainte d'optimalité (3.36b) liée au biais, on montre facilement que la solution du problème LS-SVM s'obtient par la résolution du système linéaire

$$\left[\begin{array}{c|c} 0 & \mathbb{1}^\top \\ \hline \mathbb{1} & \mathbf{K} + \frac{1}{C}\mathbf{I} \end{array}\right] \left[\begin{array}{c} b \\ \hline \alpha \end{array}\right] = \left[\begin{array}{c} 0 \\ \hline \mathbf{y} \end{array}\right] \tag{3.38}$$

avec $\mathbf{I}$, la matrice identité de dimension N.

Une fois les paramètres α et b déterminés, le modèle recherché prendre la forme $f(\mathbf{x}) = \sum_j \alpha_j K(\mathbf{x}, \mathbf{x}_j) + b$. On peut constater que tous les points dont l'erreur est non nulle correspondent à un α_i non nul. Le choix d'un coût quadratique réduit donc la parcimonie du modèle obtenu comparé à un SVR. Ce phénomène est amplifié en présence de bruit où la possibilité d'obtenir une prédiction parfaite est plus faible et de plus, non souhaitable. Néanmoins, précisons qu'une version parcimonieuse des LS-SVM a été proposée par les auteurs de cette méthode et dont les détails sont exposés dans [SLV00]. Le principe de la méthode consiste à apprendre un LS-SVM classique puis à retirer de l'ensemble d'apprentissage les données dont les poids α_i sont faibles et à recommencer l'apprentissage. On itère ce processus jusqu'à ce que l'augmentation de l'erreur d'apprentissage soit jugée trop importante.

3.6.6 Extension des méthodes à noyaux aux systèmes dynamiques

3.6.6.1 Cas des modèles à erreur d'équation

Comme nous l'avons vu précédemment, tant que le vecteur de régression ne contient que des données qui sont disponibles avant l'apprentissage, le cadre de la régression classique suffit pour appréhender la modélisation.

C'est ainsi le cas des modèles FIR ou ARX. Dans ce cadre les méthodes SVR et LS-SVM peuvent être appliquées directement sans adaptation particulière pour les systèmes dynamiques. Quelques exemples d'applications des méthodes SVR à l'identification des systèmes dynamiques peuvent être trouvés dans les références [GDH+01, YFL05, MRRÁCV+06, Lau08, MS09]. En ce qui concerne les méthodes de type LS-SVM, on peut citer des références comme [SVM01, GPSM05, ESM05, FPSM09].

3.6.6.2 Cas des modèles à erreur de sortie (OE)

Dès que le vecteur de régression fait apparaître la prédiction du modèle aux instants précédents, directement ou indirectement par le biais de l'erreur de régression, le problème devient plus complexe. Un exemple d'utilisation des SVM dans ce cadre peut être trouvé dans [Suy00]. La méthode décrite se fonde sur l'utilisation de l'approche LS-SVM vue précédemment. Le modèle recherché est de la forme : $\hat{y}_k = f(\hat{\mathbf{Y}}_{k-n_{\hat{y}}|k-1}, \mathbf{U}_{k-n_u|k-1})$ (voir l'équation 3.14 pour un rappel des définitions des vecteurs $\hat{\mathbf{Y}}$ et $\mathbf{U}$). Le principal problème provient de la récursion sur la sortie prédite. Pour simplifier la présentation et sans perte de généralités, nous allons simplement étudier les modèles de la forme $\hat{y}_k = f(\hat{\mathbf{Y}}_{k-n_{\hat{y}}|k-1})$ car le vecteur des entrées $\mathbf{U}_{k-n_u|k-1}$ n'influence pas la difficulté d'estimation des paramètres évoquée.

En s'appuyant sur un LS-SVM, alors le modèle recherché peut être écrit comme : $\hat{y}_k = \left\langle \mathbf{w}, \phi(\hat{\mathbf{Y}}_{k-n_{\hat{y}}|k-1}) \right\rangle + b$. On rappelle que l'erreur de régression est $e_k = y_k - \hat{y}_k$. La formulation du problème d'apprentissage par les LS-SVM peut alors se mettre sous la forme :

$$\begin{aligned} \min_{\mathbf{w},b,\mathbf{e}} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + \frac{C}{2}\sum_k e_k^2 \\ \text{s.c.} \quad & y_k - \left(\left\langle \mathbf{w}, \phi(\hat{\mathbf{Y}}_{k-n_{\hat{y}}|k-1}) \right\rangle + b \right) = e_k, \quad \forall k \end{aligned}$$

Le lagrangien d'un tel problème est :

$$\mathscr{L}(\mathbf{w}, b, \mathbf{e}, \alpha) = \frac{1}{2}\|\mathbf{w}\|^2 + \frac{C}{2}\sum_k e_k^2 + \sum_k \alpha_k \left[y_k - \left(\left\langle \mathbf{w}, \phi(\hat{\mathbf{Y}}_{k-n_{\hat{y}}|k-1}) \right\rangle + b \right) - e_k \right]$$

En dérivant le lagrangien par rapport aux variables primales, on obtient [Suy00] :

$$\begin{aligned} \nabla_{\mathbf{w}}\mathscr{L} &= \mathbf{w} - \sum_k \alpha_k \phi(\hat{y}_{k-1}, \hat{y}_{k-2}, ..., \hat{y}_{k-n_{\hat{y}}}) = 0 \\ \nabla_b \mathscr{L} &= \sum_k \alpha_k = 0 \\ \nabla_{e_k}\mathscr{L} &= Ce_k - \alpha_k - \sum_{i=1}^{n_y} \alpha_{k+i} \frac{\partial}{\partial e_{k-i}} \left(\left\langle \mathbf{w}, \phi(\hat{\mathbf{Y}}_{k-n_{\hat{y}}|k-1}) \right\rangle + b \right) = 0 \end{aligned}$$

En remplaçant $\mathbf{w}$ dans la dernière expression, la dernière condition d'optimalité s'écrit :

$$Ce_k - \alpha_k - \sum_{i=1}^{n_y} \alpha_{k+i} \frac{\partial}{\partial e_{k-i}} \left(\sum_l \alpha_l K(\hat{\mathbf{Y}}_{k-n_{\hat{y}}|k-1}, \hat{\mathbf{Y}}_{l-n_{\hat{y}}|l-1} \right) = 0 \tag{3.39}$$

En l'état, cette condition n'étant pas linéaire, la solution de ces équations est très difficile à obtenir car l'obtention d'un problème dual facile à résoudre devient extrêmement compliquée. La simplification proposée consiste à prendre $C \to \infty$ ce qui nous conduit à résoudre :

$$\begin{aligned} \min_{\mathbf{w},\alpha,b} \quad & \frac{1}{2}\sum_k e_k^2 \\ \text{s.c.} \quad & \sum_k \alpha_k = 0 \\ & y_k - \left(\sum_l \alpha_l K(\hat{\mathbf{Y}}_{k-n_{\hat{y}}|k-1}, \hat{\mathbf{Y}}_{l-n_{\hat{y}}|l-1}) + b \right) = e_k, \ \forall k \end{aligned} \tag{3.40}$$

Le problème d'optimisation à résoudre pour obtenir l'estimation des paramètres est non-convexe. Il peut être résolu à l'aide de méthodes d'optimisation non-linéaires. Toutefois, le cas $C \to \infty$ ne tient pas compte de la régularisation incluse dans les algorithmes SVM traditionnels. Afin d'éviter les problèmes de sur-apprentissage, cette régularisation est remplacée par une méthode d'arrêt prématuré (*early stopping*). En cela, cette démarche se rapproche de la procédure de régularisation usitée dans les réseaux de neurones. Dans la suite, cette variante OE de l'algorithme des Least Squares SVM sera dénommée par RLSSVM (Recurrent LS-SVM).

3.6.7 Problématique d'identification des modèles dynamiques basés sur les méthodes à noyaux

Dans le contexte de notre application, les choix à effectuer concernent :

- la dynamique du système à savoir l'ordre n_y (cas des modèles de type erreur d'équation i.e. ARX) ou $n_{\hat{y}}$ (modèles OE) et l'ordre n_u sur les entrées. Toutefois, pour simplifier la complexité du problème, comme indiqué dans la section relative aux résultats des modèles linéaires, nous utiliserons les ordres optimaux des modèles linéaires. Nous nous focaliserons donc sur la détermination d'une extension non-linéaire des dits modèles linéaires,
- les paramètres inhérents aux algorithmes à noyaux. Ainsi, pour le LS-SVM, il est nécessaire de spécifier le paramètre de régularisation C et l'hyper-paramètre du noyau K. Pour la méthode SVR, en plus de ces paramètres, la recherche de la valeur optimale de la largeur du tube ε doit être menée. Soulignons que dans notre contexte, les considérations pratiques de l'application (erreur minimale admissible sur la prédiction de la température) peuvent faciliter le choix de ε.

Comme pour le cas linéaire, la sélection de tous ces paramètres se fait via une méthode de validation croisée. De façon similaire au cas linéaire, le critère de validation fait appel à la sortie simulée du système et n'utilise pas les sorties mesurées (qui dans notre contexte applicatif ne seront pas disponibles lors de l'exploitation des modèles). Le critère de validation prend donc la forme (3.25a) avec la sortie simulée donnée par

$$\hat{y}_k = \sum_l \alpha_l K(\varphi_l, \varphi_k) + b \quad \text{avec} \quad \varphi_m = \begin{bmatrix} \mathbf{U}^\top_{m-n_u|m-1} & \hat{\mathbf{Y}}^\top_{m-n_{\hat{y}}|m-1} \end{bmatrix}^\top, \quad m \in \{k,l\}. \tag{3.41}$$

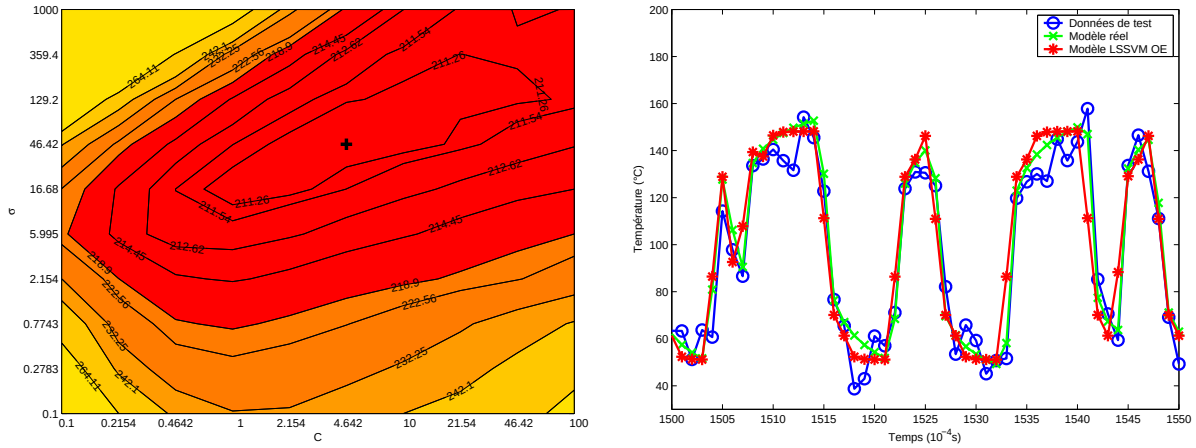
3.6.8 Application des méthodes à noyaux pour prédire la température interne des composants électroniques

Pour mettre en oeuvre les SVR avec coût ε-insensible sur les données « Transistor », nous avons utilisé la toolbox « SVM and Kernel Methods Matlab Toolbox » [CGGR05]. En revanche, la taille des données « Maquette » étant plus importante, nous avons eu recours à la toolbox LibSVM [CL01] qui implémente les machines à noyaux avec un système de mémoire cache afin de gérer la matrice des noyaux $\mathbf{K}$. Pour les LS-SVM, la toolbox « LS-SVMlab » [SGB+02] a été utilisée. Pour le RLSSVM (pour rappel, il s'agit de la version OE des LS-SVM), c'est la fonction `fmincon` de la toolbox « Optimization » de matlab qui est utilisée afin de résoudre la forme primale du problème d'optimisation sous contrainte d'égalité (3.40).

On reprend la structure des vecteurs de régression obtenus dans le cas linéaire ($n_y = 1$ et $n_u = 4$ pour les données « Transistor » et « Maquette »). Nous avons considéré une fonction noyau gaussienne usuellement utilisée pour la régression par les SVM. Pour les SVR utilisant un coût ε-insensible, la largeur ε du tube est fixée à 0.1 pour les données « Transistor » (niveau de bruit connu) et à 0.01 pour les données « Maquette » (compte tenu de la précision attendue sur ces données et du filtrage effectué) pour les données normalisées. Ces valeurs de ε correspondent respectivement à une précision de 10°C et 0.5° sur la prédiction de la température pour les données « Transistor » et « Maquette »). Les paramètres restant à choisir sont donc le paramètre de régularisation C et la largeur de bande de la fonction noyau. Pour chaque base de données, nous avons testé les trois types de modélisation à noyaux, à savoir SVR, LS-SVM et RLSSVM. Dans chaque cas, différents modèles sont élaborés, correspondant à différentes

valeurs des hyper-paramètres ((C,σ) pour SVR et LS-SVM et σ pour le RLSSVM). La qualité de chacun de ces modèles est évaluée sur la base de l'erreur quadratique moyenne en validation conformément à la procédure de validation utilisée pour l'élaboration des modèles linéaires.

Nous présentons ci-après les résultats obtenus sur les deux bases de données pour les algorithmes LS-SVM et SVR puis nous évoquerons le cas particulier de l'algorithme RLSSVM. Ainsi, sur les données « Transistor », la figure 3.16(a) résume l'évolution de l'erreur en validation en fonction de (C,σ) pour la méthode LS-SVM. On constate que le modèle optimal est atteint pour $(C,\sigma) = (4.64, 46.41)$. L'erreur de validation correspondante est de l'ordre de 210.969 et l'erreur évaluée sur les données de test vaut 203.765 et 99.324 respectivement sur les données bruitées et sans bruit. La figure 3.16(b) compare la sortie du modèle LS-SVM testé en simulation (suivant la formule de l'équation 3.41) avec les données réelles. On constate une certaine adéquation du modèle aux données.



(a) Critère de validation en fonction du couple (C,σ). L'optimum du critère de validation est matérialisé par une croix.

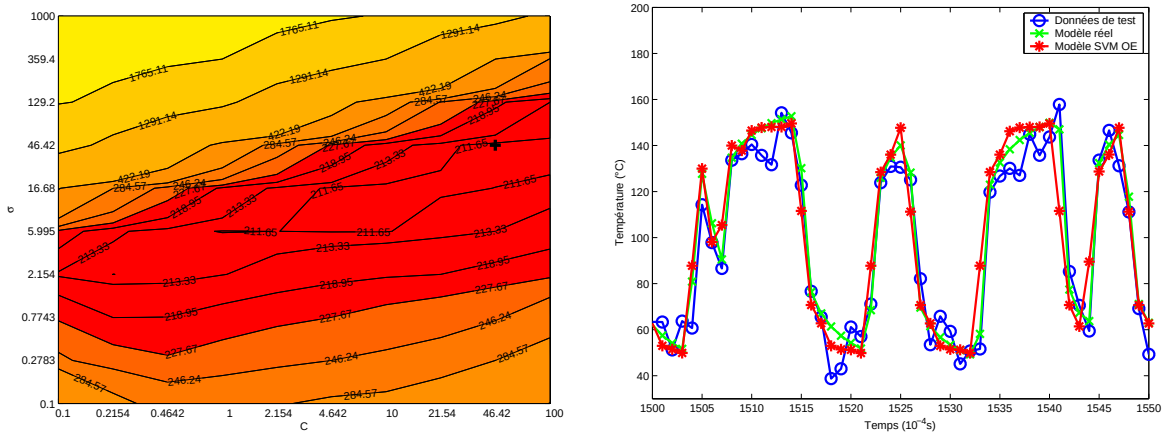
(b) Sortie du modèle optimal comparée aux données réelles (bruitées et non bruitées) sur une partie des données de test.

FIGURE 3.16 : Récapitulatif du protocole de validation et performances du meilleur modèle fourni par la méthode LS-SVM sur la base de données « Transistor ».

La figure 3.17 regroupe les résultats obtenus pour la méthode SVR. En particulier, on constate que le meilleur résultat est obtenu pour $(C,\sigma) = (46.4159, 46.4159)$, ce qui correspond à un critère de validation de l'ordre de 211.199 et une erreur de test égale à 203.292 (respectivement 98.395 sur les données sans bruit). Finalement, la sortie du modèle optimal SVR évaluée sous la forme d'un modèle OE est présentée sur la figure 3.17(b)) où on constate également une adéquation assez satisfaisante aux données réelles.

Bien que les résultats bruts des méthodes à noyaux sur les données « Transistor » aient l'air satisfaisants, ils sont en réalité moins bons que ceux obtenus avec les modèles linéaires si on compare les performances des modèles SVR et LS-SVM avec celles du tableau 3.4. L'utilisation des SVM à régression semble ne pas apporter une amélioration aux résultats de modélisation linéaire ou par réseau de neurones. Deux raisons possibles peuvent justifier ce constat : d'une part, les données « Transistor » semblent présenter un comportement dynamique linéaire (même si la génération de ces données par des méthodes numériques à éléments finis a intégré une non-linéarité liée à la variation de la conductivité thermique des matériaux en fonction de la température) ; d'autre part les modèles SVR et LS-SVM ont été identifiés sous forme de modèle à erreur d'équation (ce qui permet de préserver la convexité du problème d'optimisation) mais ont été testés en simulation. Cette dernière situation pénalise la qualité des prédictions obtenues car les modèles ont été testés dans un cadre différent de celui dans lequel ils ont été élaborés.

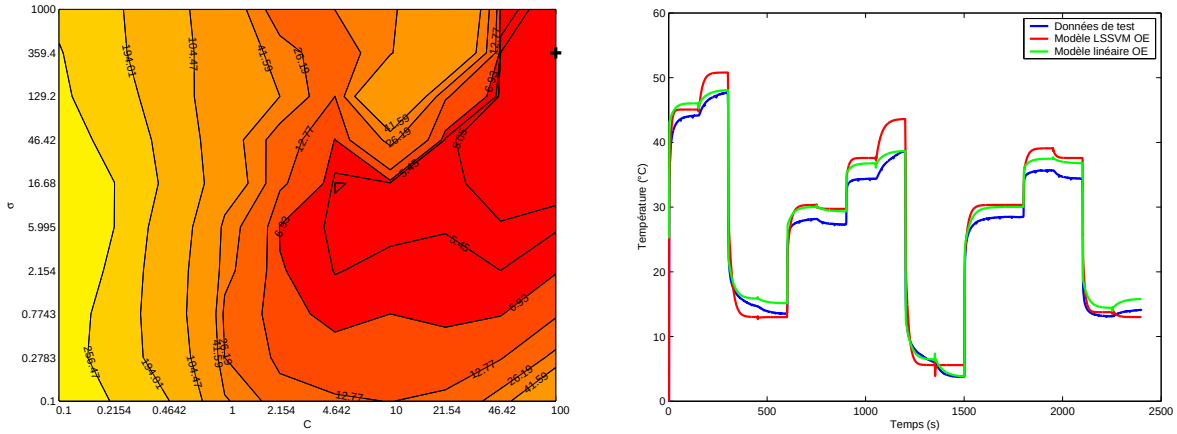
Les méthodes LS-SVM et SVR ont été ensuite testées sur les données « Maquette » en suivant le protocole expérimental usuel et les résultats obtenus sont tracés sur les graphiques des figures 3.18 et 3.19. Ainsi pour le LS-SVM, d'après la figure 3.18(a), on constate que le meilleur modèle est obtenu pour $(C,\sigma) = (100, 359.3814)$ avec



(a) Critère de validation en fonction de C, σ). (b) Sortie du modèle optimal comparée aux données réelles.

FIGURE 3.17 : Résultats obtenus par la méthode SVR sur la base de données « Transistor ».

des erreurs de validation et de test (évaluées sous forme d'erreur quadratique moyenne) qui valent respectivement 4.053 et 6.501. La sortie simulée correspondant à ce modèle est présentée sur la figure 3.18(b) où on remarque que les résultats de ce modèle non-linéaires sont moins satisfaisants que ceux du modèle linéaire OE.

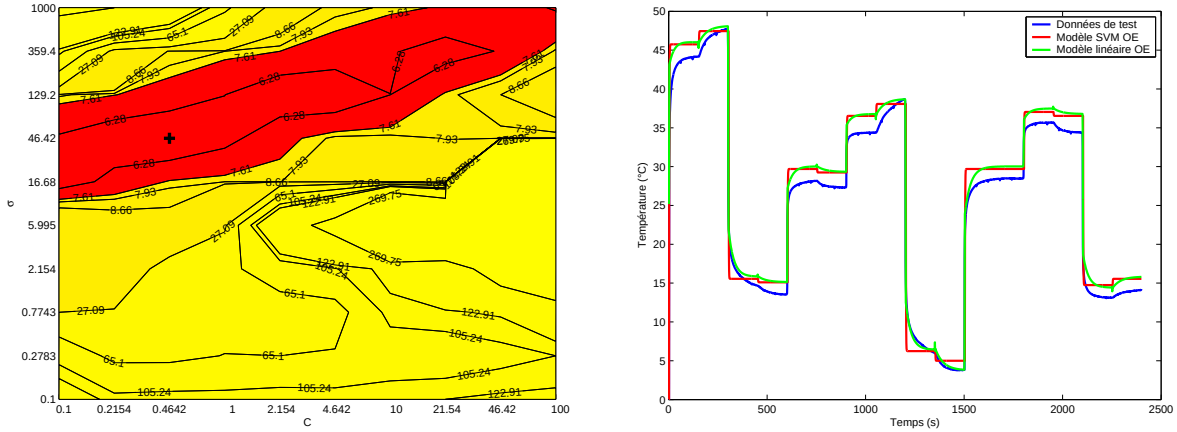


(a) Critère de validation en fonction de (C, σ). Le modèle optimal est matérialisé par une croix. (b) Comparaison des sorties du modèle LS-SVM optimal, du meilleur modèle linéaire avec les données réelles.

FIGURE 3.18 : Résultats de la modélisation de la température des composants à partir de la base de données « Maquette » en utilisant la méthode LS-SVM.

En ce qui concerne la méthode SVR, le meilleur modèle que nous en avons déduit suivant notre protocole expérimental correspond à une paire d'hyper-paramètres $(C, \sigma) = (0.4642, 46.4159)$. Ce modèle optimal donne les performances en généralisation suivantes : le critère de validation est de l'ordre de 4.816 et l'erreur de test est de 4.575. La comparaison de sa sortie avec celle du modèle linéaire correspondant est décrite à la figure 3.19(b). On constate cette fois-ci que le modèle SVR est très proche du modèle linéaire même si ses performances en généralisation sont légèrement inférieures (voir tableau 3.5 pour les résultats du modèle linéaire).

En guise de conclusion à la modélisation LS-SVM et SVR, on constate que leur utilisation n'apporte pas d'amélioration sensible aux résultats obtenus jusqu'alors (modèles linéaires ou réseaux de neurones) sur les deux bases de données à notre disposition. Pour améliorer les résultats des méthodes à noyaux, on peut envisager la recherche de la structure optimale du vecteur de régression, au lieu de se baser sur le vecteur de régression obtenu dans le cas linéaire. Précisons toutefois que cette recherche peut se révéler coûteuse en termes de temps de calcul. Un autre point à souligner est que les modèles LS-SVM semblent sur-apprendre comparativement aux modèles



(a) Critère de validation en fonction de (C, σ).

(b) Sortie du modèle SVR comparée aux données réelles et à la sortie du meilleur modèle linéaire OE.

FIGURE 3.19 : Résultats de la modélisation de la température des composants à partir de la base de données « Maquette » en utilisant la méthode SVR (coût ε-insensible).

SVR. En effet, en comparant les performances en validation et en test des deux techniques à noyaux, on remarque que les modèles LS-SVM tendent à fournir une erreur de validation plus faible par rapport aux SVR mais en revanche leurs erreurs de test sont sensiblement plus élevées.

Une raison évoquée pour justifier les résultats des méthodes à noyaux est que ces modèles sont identifiés en erreur d'équation (ils utilisent un vecteur de régression de type 3.19) mais sont évalués en erreur de sortie (voir l'équation 3.41). Une solution est donc d'appliquer la méthode du RLSSVM qui optimise directement le modèle en erreur de sortie quitte à perdre la convexité du problème. Le test de cette méthode a été fait sur les données « Transistor ». Compte tenu des temps de calcul prohibitifs de la méthode, nous n'avons pas conduit une recherche optimale de la largeur de bande σ du noyau gaussien (étant entendu que le paramètre de régularisation est fixé à $C = \infty$ pour le RLSSVM) mais nous avons considéré la valeur optimale de ce paramètre fournie par nos expériences sur la méthode LS-SVM. Ce choix semble raisonnable dans la mesure où l'initialisation du RLSSVM se fait à partir de la solution LS-SVM.

Pour un nombre maximal d'itérations de l'algorithme d'optimisation fixé à 20, on obtient les résultats décrits sur les figures 3.20 et 3.21. L'erreur de test sur les données « Transistor » est de 241.580 (138.305 sur les données sans bruit) et de 153.280 sur les données « Maquette ». L'utilisation de cette méthode n'apporte donc pas d'amélioration vis-à-vis des méthodes précédentes et notamment vis-à-vis du modèle LSSVM identifié en ARX. Pour expliquer ce phénomène, il convient de noter que la régularisation par « early-stopping » est beaucoup moins fiable théoriquement que celle utilisée pour les LSSVM.

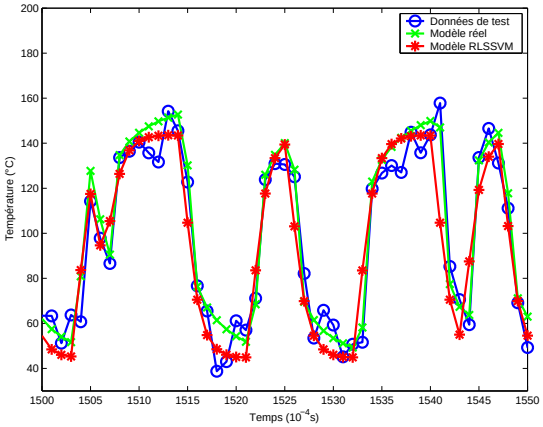


FIGURE 3.20 : Sortie du modèle OE pour un RLSSVM comparées aux données réelles sur la base de données « Transistor ».

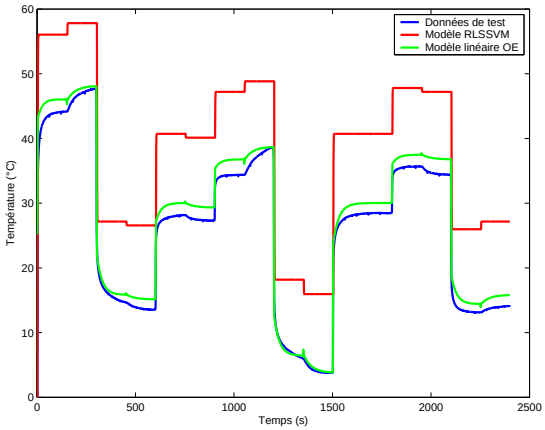


FIGURE 3.21 : Sortie du modèle OE pour un RLSSVM comparées aux données réelles sur la base de données « Maquette ».

3.7 Modélisation par réseaux bayésiens

3.7.1 Définitions

3.7.1.1 Réseaux bayésiens

Un réseaux bayésien est (figure 3.22) défini par ([NWL+04]) :
- un graphe acyclique orienté G, $G = (V, E)$, où V est l'ensemble des noeuds de G, et E l'ensemble des arcs de G,
- un espace probabilisé fini (Ω, Z, p),
- un ensemble de variables aléatoires associées aux noeuds du graphe et définies sur (Ω, Z, p), tel que : $P(V_1, V_2, ..., V_n) = \prod_{i=1}^{n} P(V_i|C(V_i))$ où $C(V_i)$ est l'ensemble des causes (parents) de V_i dans le graphe G et n est le nombre de noeuds de G.

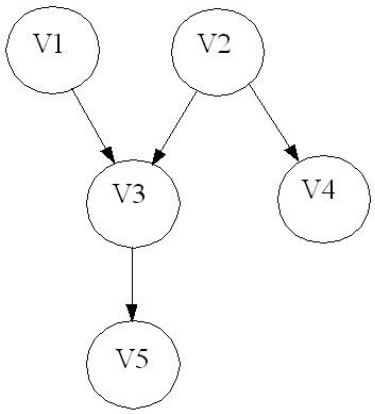


FIGURE 3.22 : Exemple de réseau bayésien. La loi jointe $P(V_1, V_2, V_3, V_4, V_5)$ peut alors être réécrite sous la forme $P(V_1) \times P(V_2) \times P(V_3|V_1, V_2) \times P(V_4|V_2) \times P(V_5|V_3)$

3.7.1.2 Indépendance conditionnelle

Les réseaux bayésiens permettent une réécriture simplifiée de lois jointes. Tout d'abord, les arcs du graphe représentent une dépendance entre des variables. De plus, le graphe apporte également de l'information sur l'indépendance conditionnelle entre ces variables. En effet, les parents du noeud V sont définis comme les noeuds ayant un arc en commun avec V et qui pointe vers V, et inversement pour les enfants de V. Par extension, on définit également la descendance de V. On peut alors établir que toute variable d'un réseau bayésien est indépendante de ses non-descendants conditionnellement à ses parents.

On peut généraliser ces indépendances conditionnelles à des ensembles de variables. Pour cela, on utilise le concept de d-séparation. Soient S_1, S_2 et S_3 trois ensembles de variables appartenant à un réseau bayésien. On dit que l'ensemble S_3 d-sépare les ensembles S_1 et S_2 si pour tout chemin reliant S_1 à S_2, il existe une variable V telle que :

- soit $V \in S_3$,
- soit $V \notin S_3$ et V n'a pas d'enfant.

Si un ensemble d-sépare S_1 et S_2, alors S_1 et S_2 sont conditionnellement indépendants.

3.7.2 Les différents réseaux bayésiens

Les variables aléatoires présentes dans un réseau bayésien peuvent être de deux types en fonction des valeurs qu'elles peuvent prendre. Si ce nombre de valeurs est fini, la variable est dite « discrète » et la variable suit alors une loi multinomiale. Pour les variables à valeur continue, plusieurs lois sont possibles. Toutefois, dans le cadre des réseaux bayésiens étudiés dans cette thèse, les variables continues suivent uniquement des lois gaussiennes.

3.7.2.1 Réseau bayésien statique

Pour ce type de réseau, le seul élément à définir est la manière de calculer les probabilités conditionnelles d'une variable connaissant ses parents. Compte tenu des différents types de variables, plusieurs cas sont possibles.

Réseau bayésien multinomial

Si toutes les variables V_i sont discrètes et suivent une loi multinomiale, le réseau est dit multinomial. Ses fonctions de probabilité sont définies à partir de tableaux de probabilité conditionnelle qui définissent la probabilité de chaque état possible pour la variable en fonction des valeurs possibles pour ses parents.

Réseau bayésien multinormal

Dans un réseau multinormal, toutes les variables V_i sont continues et suivent une loi normale, définie par un vecteur des moyennes μ de dimension d et Σ la matrice de covariance :

$$\mathcal{N}(x,\mu,\Sigma) = \frac{1}{(2\pi)^{\frac{d}{2}}|\Sigma|^{\frac{1}{2}}} exp\left(-\frac{1}{2}(x-\mu)^{\top}\Sigma^{-1}(x-\mu)\right)$$

La fonction de densité pour la variable V_i est alors définie par un produit des fonctions de probabilité conditionnelle :

$$f(V_i|C(V_i)) \sim \mathcal{N}(V_i, \mu_i + \sum_{V_j \in C(V_i)} \beta_{i,j}(v_j - \mu_j), \nu_i)$$

où v_j représente la valeur prise par V_j et $\beta_{i,j}$ est le coefficient de régression entre V_i et son parent V_j.

Modèle de mélange gaussien

Pour un lien d'une variable discrète vers une variable continue, une paire de paramètres espérance/variance est définie pour chaque valeur possible de la variable parente. Cette définition permet de rapprocher ce type de lien d'un modèle de mélange gaussien, qui est une combinaison linéaire de fonctions gaussiennes. La densité conditionnelle d'une variable x est alors calculée comme la somme pondérée de N gaussiennes. On a alors :

$$f(x) = \sum_{i=1}^{N} w_i \mathcal{N}(x,\mu_i,\Sigma_i) \text{ avec } \sum_{i=1}^{N} w_i = 1 \text{ et } \forall i,\ 0 \leq w_i \leq 1$$

Ce modèle peut alors être représenté par un réseau bayésien intégrant les deux types de variables, une variable discrète à N valeurs, dont la table de probabilité conditionnelle définit le poids de chaque gaussienne dans le mélange, et une variable continue, qui définit les paramètres de chacune des lois du mélange.

Fonction softmax

Pour un lien d'une variable continue vers une variable discrète, il est possible d'utiliser une fonction de type « softmax » pour définir les probabilités conditionnelles. La probabilité conditionnelle de la variable est alors définie par :

$$P(V_i = i|V_j = x) = \frac{exp\,(a_i x + b_i)}{\sum_k exp\,(a_k x + b_k)}$$

où V_i est une variable discrète, V_j une variable continue et les a_k et b_k ($k \in [1,N]$ où N est le nombre de valeurs prises par V_i) sont les paramètres de la fonction « softmax »

3.7.2.2 Réseaux bayésiens dynamiques

Les réseaux bayésiens temporels (ou Dynamic Bayesian Network, DBN) sont une extension des réseaux bayésiens précédents servant à modéliser l'évolution temporelle des différentes variables. Leur structure repose sur un réseau bayésien classique, mais une même grandeur à différents instants va être représentée par plusieurs variables dans le réseau. L'ensemble des variables appartenant à un instant est appelé « slice ». La structure du réseau bayésien dans une slice définit les liens *intra-slices*. Pour permettre une modélisation dynamique, on autorise des arcs *inter-slices* qui vont relier 2 variables appartenant à 2 slices différents. Pour conserver des modèles causaux, les liens inter-slices se font toujours des instants passés vers les instants futurs. De plus, on considère le plus souvent que les réseaux bayésiens répondent à une condition de Markov d'ordre 1, c'est-à-dire que pour une slice donnée, les liens inter-slices ne peuvent provenir que de la slice précédente.

Les réseaux bayésiens temporels permettent également une homogénéisation de l'écriture des différents modèles graphiques dynamiques existants ([Mur02]). Ainsi, leur formalisme très ouvert regroupe sous une même notation les modèles de Markov cachés (ou Hidden Markov Model HMM) et leurs dérivés (figure 3.23), les filtres de Kalman mais aussi des modèles beaucoup plus complexes.

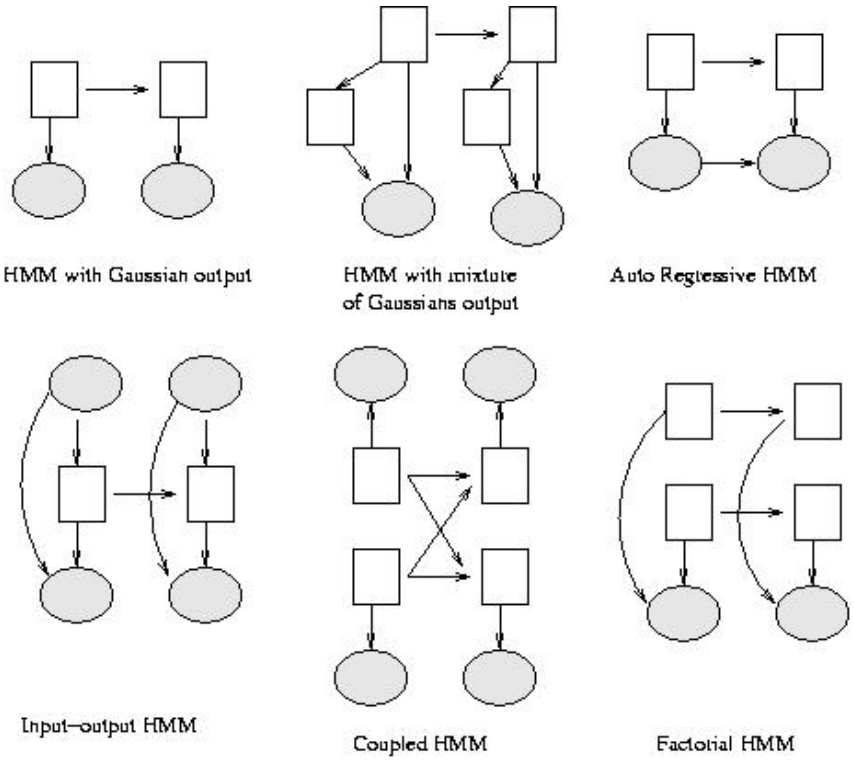


FIGURE 3.23 : Représentation des différentes formes de HMM sous un formalisme de DBN. Les variables observées sont grisées. Les variables discrètes sont mises sous forme de rectangles et les variables continues sous forme de ronds.[Mur02]

3.7.3 Inférence

L'inférence consiste, à partir d'une structure connue, à calculer les probabilités marginales a posteriori de quelques variables sachant la valeur des variables observées.

3.7.3.1 Inférence exacte

Soient O l'ensemble des variables observées et X l'ensemble des variables dont on cherche à établir les probabilités conditionnelles. L'observation $O = o$ est souvent appelée évidence. Le but de l'inférence est alors de calculer $P(X|O = o)$. Le calcul de la probabilité $P(X|O = o)$ revient à marginaliser la loi jointe associée au réseaux bayésiens pour la ou les variables concernées.

Bucket elimination

En ce sens, l'utilisation d'un réseau bayésien présente déjà une utilité. Toutefois, il peut être utile d'organiser intelligemment les calculs nécessaires afin de ne pas avoir à effectuer plusieurs fois les mêmes opérations. Ce principe guide la méthode dite de *bucket elimination* [Dec96]. L'idée est ici d'éliminer une par une toutes les variables qui ne sont pas incluses dans X en regroupant toutes les expressions où elles sont présentes puis en les marginalisant. En procédant ainsi, il est possible de limiter le nombre d'opérations nécessaires et d'obtenir le résultat recherché.

Ainsi, dans le réseau présenté dans la figure 3.22, on a :

$$P(V_1,V_2,V_3,V_4,V_5) = P(V_1)P(V_2)P(V_3|V_1,V_2)P(V_4|V_2)P(V_5|V_3).$$

De plus, quelque soit l'évidence $O = o$, on sait que :

$$P(V_1,V_2,V_3,V_4,V_5|O = o) = \frac{P(V_1,V_2,V_3,V_4,V_5,O = o)}{P(O = o)}.$$

Ainsi, si l'on cherche à calculer la probabilité de la variable V_5 sachant que la variable V_2 prend la valeur v_2, il est possible d'organiser les calculs de la manière suivante :

$$\begin{aligned} P(V_5,V_2 = v_2) &= \sum_{V_1,V_3,V_4} P(V_1,V_2 = v_2,V_3,V_4,V_5) \\ &= P(V_5|V_3)P(V_2 = v_2)\sum_{V_1}P(V_1)\sum_{V_3}P(V_3|V_1,V_2 = v_2)\sum_{V_4}P(V_4|V_2 = v_2). \end{aligned}$$

Dans cet exemple, la marginalisation sur la variable V_4 est ici très simple puisque $\sum_{V_4} P(V_4|V_2 = v_2) = 1$. En éliminant ainsi successivement les variables, on obtient au final la probabilité recherchée. La structure du réseau a donc permis de réduire les calculs nécessaires.

Message passing

En exploitant la structure du réseau et la programmation dynamique, il est possible de développer un algorithme d'inférence très efficace. Pour appliquer l'algorithme de *message passing*[Pea88], on considère le cas d'une structure de réseau en polyarbre (arbre à plusieurs racines). En effet, dans ce type de structure, la d-séparation s'applique de manière très simple. Chaque noeud X d-sépare ses descendants de ses non-descendants. De plus, chaque variable est d-séparée de ses frères conditionnellement à ses parents et aussi à chacun des parents de ses fils conditionnellement à ses fils. Enfin, il n'existe qu'un seul chemin entre 2 variables. Chaque noeud X d-sépare donc le polyarbre en 2 sous-polyarbres indépendants conditionnellement. Or, l'évidence $O = o$ peut être présente dans chacun des 2 polyarbres. On peut donc distinguer l'évidence qui va parvenir au noeud via ses parents, notée o_X^+, et l'évidence qui va parvenir au noeud via ses enfants, notée o_X^-. La probabilité conditionnelle au noeud considéré va alors être :

$$\begin{aligned} P(X|O = o) &= P(X|o_X^+,o_X^-) \\ &= \frac{1}{P(o_X^+,o_X^-)}P(o_X^+,o_X^-|X)P(X) \end{aligned}$$

o_X^+ et o_X^- étant indépendantes sachant X, on peut écrire :

$$\begin{aligned} P(X|O=o) &= \frac{1}{P(o_X^+,o_X^-)}P(o_X^+|X)P(o_X^-|X)P(X) \\ &= \frac{1}{P(o_X^+,o_X^-)}P(o_X^+|X)P(o_X^-,X) \\ &\propto P(o_X^-|X)P(o_X^+,X). \end{aligned}$$

On définit alors deux termes $\lambda(X) = P(o_X^-|X)$ et $\pi(X) = P(o_X^+,X)$, que l'on cherche à calculer pour chacun des noeuds. On considère que X possède p parents $P = P_1,...,P_p$ et f enfants $F = F_1,...,F_f$. On peut alors écrire en sachant que X d-sépare ses enfants :

$$\begin{aligned} \lambda(X) &= P(o_{X,F_1}^-,...,o_{X,F_f}^-|X) \\ &= \prod_{i=1}^{p} P(o_{X,F_i}^-|X) \\ &= \prod_{i=1}^{p} \lambda_{F_i}(X). \end{aligned}$$

En utilisant le principe de d-séparation pour les parents de X, on peut également établir :

$$\pi(X) = \sum_{o_P} P(X|P=o_P,o_X^+)\prod_{i=1}^{p}\pi_X(o_{p_i}).$$

Ce raisonnement est vrai pour chacune des variables du réseau. En initialisant les messages à partir des noeuds feuilles, des noeuds racines et des observations, chaque variable va pouvoir recevoir les messages nécessaires puis envoyer à son tour un message aux noeuds voisins. À la fin de l'algorithme, l'évidence a été propagée à l'ensemble du réseau.

Pour étendre l'algorithme de *message passing* aux structures qui ne sont pas des arbres, on définit l'arbre de jonction du réseau [JLO90]. Cet arbre est obtenu par les étapes de moralisation (mariage des parents des noeuds) et de triangulation (les cycles de plus de 4 noeuds sont cassés par l'ajout d'un arc). Dans cet arbre, les noeuds appartenant à un sous-graphe complètement connecté sont regroupés au sein de cliques, qui sont par la suite traitées comme des variables au sein de l'arbre. Pour revenir aux probabilités des variables originelles, on effectue ensuite l'inférence localement au sein de la clique.

Une méthode pour appliquer les algorithmes classiques d'inférence sur les réseaux bayésiens dynamiques consiste à « déplier » temporellement le réseau. Toutefois, le réseau obtenu possède alors des cliques de très grandes tailles. Cependant, compte tenu du fait que les noeuds d'une « slice » d-sépare les slices passées des slices futures, on peut réduire l'inférence à un réseau représentant deux slices successives, en iérant le processus le nombre de fois nécessaires selon la longueur temporelle de l'évidence traitée. C'est le principe du Frontier Algorithm qui ne retient que les noeuds cachés de la slice précédente pour effectuer l'inférence dans la slice courante. Si l'on applique cet algorithme de propagation sur un réseau bayésien dynamique de type HMM, on retrouve l'algorithme *Forward-Backward* usuel. Pour des noeuds cachés discret, ceci s'apparente à du filtrage de Kalman. On peut réduire le nombre de noeuds retenus dans la slice précedente pour obtenir l'Interface Algorithm de Murphy. Les justifications et les détails d'implémentation des différentes méthodes sont tous précisés dans le travail de Murphy [Mur02].

3.7.3.2 Inférence approchée

Loopy Belief Propagation

L'algorithme de *message passing* présente cependant un inconvénient. En effet, même si cet algorithme est exacte, il reste très complexe et l'arbre de jonction peut contenir des cliques de très grande taille. On peut alors recourir à des méthodes d'inférence approchée, telle que le *Loopy Belief Propagation*, où l'algorithme de *message passing* est appliqué directement au réseau bayésien, même si celui-ci ne possède pas une structure de polyarbre.

3.7.4 Apprentissage

3.7.4.1 Apprentissage des paramètres sur données complètes

Si l'on suppose la structure du réseau connue et que les évidences couvrent l'ensemble des variables du réseau, l'apprentissage des paramètres est effectué en utilisant des méthodes telles que le maximum de vraisemblance. Soit w l'ensemble des paramètres du réseau bayésien. Ainsi, si toutes les variables sont observées (on appelle v l'ensemble de ces évidences v_i), il est possible de calculer la *log-vraissemblance* du réseau, définie par :

$$\mathbf{L}(w) = \log P(V = v|w) = \log \prod_i P(v_i|w) = \sum_i \log P(v_i|w)$$

Les paramètres optimaux w^* sont alors naturellement calculés par :

$$w^* = \text{argmax}_w \mathbf{L}(w)$$

3.7.4.2 Apprentissage des paramètres sur données incomplètes

Si des variables sont exclues des évidences, on dit alors qu'elles sont cachées. Il est alors nécessaire d'utiliser un algorithme de type *Expectation Minimization*. On suppose que l'on connaît un algorithme d'inférence pour le réseau bayésien dont on cherche à estimer les paramètres. Soit w les paramètres, V_o les variables observées (v_o leur valeur) et V_c les variables cachées.

Pour initialiser l'algorithme, il faut fixer a priori les paramètres initiaux w^0 du réseau. Il est alors possible d'inférer une première fois la valeur des variables cachées. C'est la phase d'*expectation*. La distribution $\mathbf{D}$ des variables cachées est alors donnée par :

$$\mathbf{D} = P(V_c|V_o = v_o, w)$$

Une fois que l'on connait la distribution des variables cachées, il est possible d'utiliser une nouvelle fois l'approche de maximisation de la vraisemblance. La log-vraisemblance est :

$$\mathbf{L}(w) = \log P(V_o = v_o|w) = \log \sum_{V_c} P(V_o = v_o, V_c|w)$$

ce qui correspond la phase de *maximalisation*.

Ces deux phases E (*expectation*) et M (*maximalisation*) sont alors itérées jusqu'à convergence des paramètres (l'annexe D détaille les justifications de cette méthode).

Pour les réseaux bayésiens dynamiques, si l'on dispose d'un algorithme d'inférence, EM s'applique comme pour une réseau bayésien classique. On peut noter que l'application de l'algorithme EM dans un HMM est notamment équivalente à l'application de l'algorithme **Baum-Welch**, déjà connu pour ce type de structures.

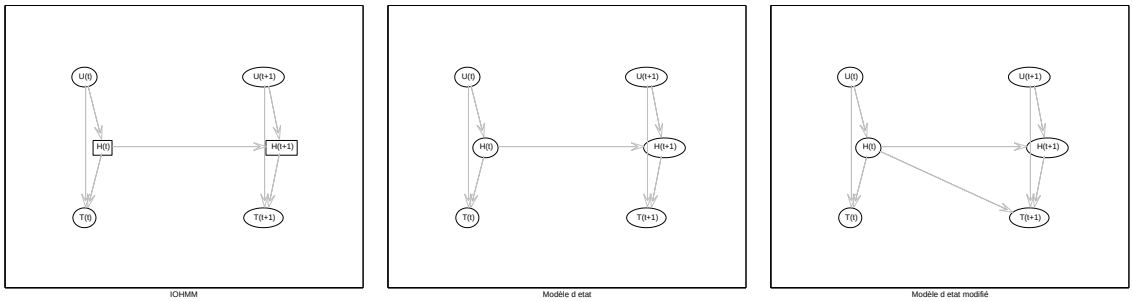
3.7.4.3 Apprentissage de structure

L'apprentissage de la structure du réseau peut être résolue par deux approches. La première consiste à effectuer des tests de dépendance et d'indépendance entre les différentes variables pour placer les arcs entre les noeuds.

La seconde approche utilise les fonctions de score, telles que le score BIC (voir tableau 3.2), afin d'évaluer la pertinence de la structure. Compte tenu de l'espace de recherche extrêmement important, des méthodes heuristiques sont souvent utilisées pour limiter les modifications effectuées à partir d'une structure initiale définie a priori. Toutefois, l'application des méthodes d'apprentissage de structure pour les réseaux dynamiques n'ont pas été utilisées dans notre application, du fait de la lourdeur de ces algorithmes et de la relative simplicité de nos données.

3.7.5 Application

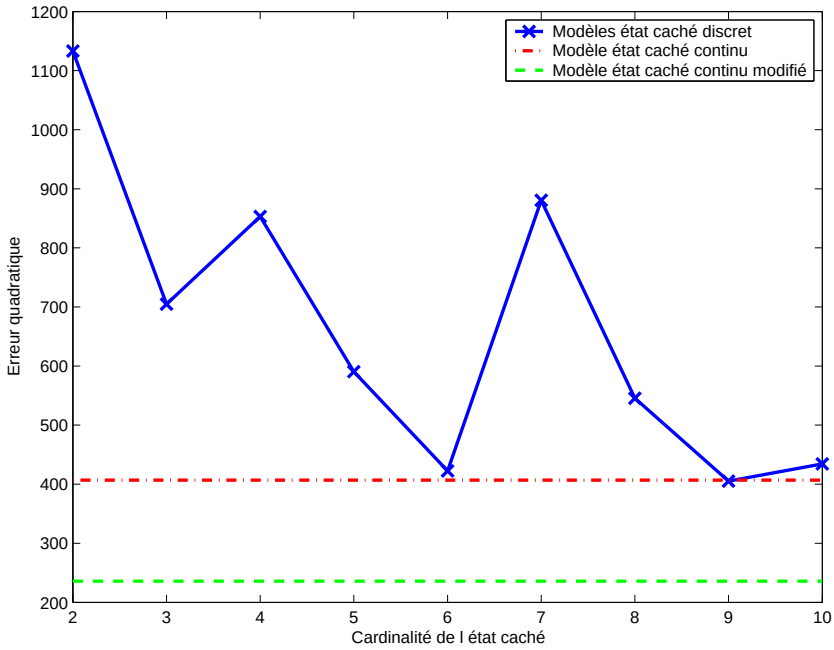
Pour notre application, le réseau utilisé doit disposer d'entrées et de sorties. Ce type de problème est traditionnellement traité par un IOHMM dans le cas des réseaux bayésiens. La structure du modèle est alors celle présentée dans la figure 3.24 (a) avec un état caché discret. Nous avons également traité à titre de comparaison le cas où la variable cachée est continue (3.24 (b)), ce qui s'apparente aux modèles d'états présentés dans le chapitre suivant. Toutefois, dans le cas où toutes les variables sont continues, on sait que le modèle est linéaire. Sans modification, le modèle est donc équivalent à un modèle OE testé précédemment. Aussi, en utilisant la flexibilité des réseaux bayésiens, le choix s'est porté aussi sur un architecture légèrement différente présenté sur la figure 3.24 (c). Dans tous les cas, l'algorithme Interface a été utilisé.

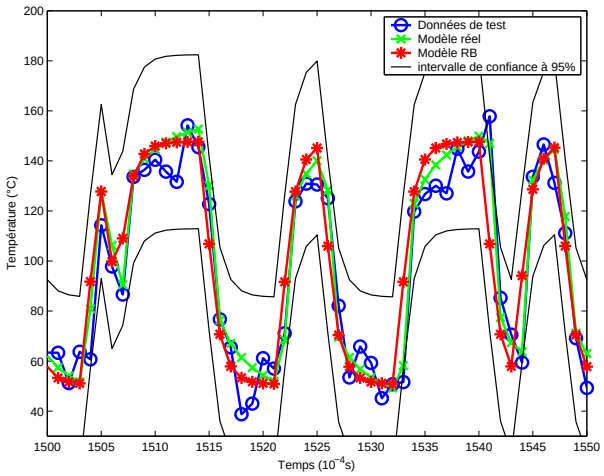


(a) IOHMM (b) Représentation d'état (c) Représentation d'état

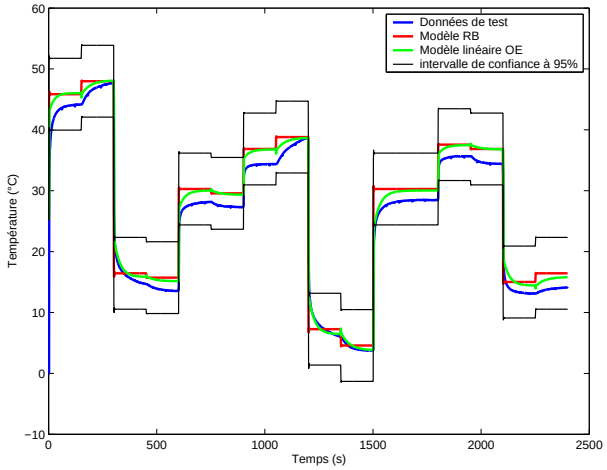
FIGURE 3.24 : Différentes structures testés dans le cadre des réseaux bayésiens. Les variables discrètes sont représentées sous forme de rectangle et les variables continues sont ovales. (U représente l'entrée du modèle, H l'état caché et T la sortie).

La structure étant fixée, dans le cas du IOHMM, le seul paramètre est la cardinalité de l'état caché. La cardinalité de l'état caché est donc testée par la même procédure de validation croisée que pour les hyperparamètres les autres méthodes d'apprentissage. Pour les deux cas où l'état caché est continu, on supposera que cette cardinatlié est de 1 (Pour le cas où plusieurs plusieurs états cachés sont présents, il est conseillé d'utiliser les méthodes développé dans le chapitre suivant). Pour les données « Transistor », on obtient la figure 3.7.5. Le meilleur modèle est donc celui de la représentation d'état modifié. L'erreur de test de ce modèle est alors de 209.803 pour les données avec bruit et 104.580 pour les données sans bruit. La sortie du modèle est comparée aux données de test sur la figure 3.25(a). Pour les données « Maquette », les modèles IOHMM n'ont pu être comparés. En effet, même s'ils ont été appris, un problème de stabilité numérique empêche l'inférence sur la base de test. Les erreurs de validation des 2 modèles de représentation d'état ont une erreur quadratique équivalente. Notre choix s'est porté sur la représentation d'état modifiée. L'erreur de test est alors de 7.536. La sortie du modèle est comparée aux données de test sur la figure 3.25(b). Les résultats pour les deux jeux de données sont moins bons que ceux les modèles linéaires, réseaux de neurones et même que ceux des support vector machines. La condition de Markov d'ordre 1 est sans doute trop restrictive pour obtenir un modèle performant sur ces données. L'ajout de variables « retard » pour augmenter la fenêtre temporelle sur l'entrée ou l'augmentation du nombre d'états cachés pourrait permettre d'améliorer ces résultats.





(a) Sortie obtenue pour un réseau bayésien dynamique sur les données « Transistor ».



(b) Sortie obtenue pour un réseau bayésien dynamique sur les données « Maquette ».

3.8 Conclusion

Ce chapitre nous a permis de tester différentes approches boîte noire d'identification de modèles dynamiques explicatifs des variations de la température interne des composants. Les modèles testés sont : modèles linéaires, réseaux de neurones récurrents, méthodes à noyaux pour la régression et réseaux bayésiens. Ces tests ont été conduits sur les bases de données « Transistor » et « Maquette » représentatives respectivement des dynamiques rapides et longues du phénomène thermique étudié. Cette conclusion dresse un bilan récapitulatif et comparatif des résultats obtenus.

Pour les données « Transistor », les résultats sont résumés dans les tables 3.8 et 3.9. Précisons que les performances en test présentées dans le tableau 3.9 sont issues de l'évaluation des différents modèles obtenus sur les données « Transistor » non bruitées i.e. les données brutes issues de la simulation par éléments finis. Les deux tableaux montrent que les meilleurs résultats sont obtenus par les réseaux de neurones récurrents suivis de très près par les modèles linéaires en particulier le modèle linéaire OE. Ces constats, en plus du fait que le modèle neuronal optimal comporte un seul neurone et les méthodes à noyaux pour la régression sont peu performantes, suggèrent que les données « Transistor » peuvent être décrites par un modèle essentiellement linéaire. De surcroit, l'erreur qui subsiste pour les modèles linéaires semble être plus liée à un échantillonnage non régulier lors de la génération des données qu'à une mauvaise identification du modèle (voir figure 3.4). Au delà de ces aspects, on peut noter aussi que le modèle OE (linéaire ou réseau de neurones) fournit en test une erreur absolue moyenne (dernière colonne du tableau 3.8) qui est de l'ordre de la variance du bruit affectant les données (voir figure 1.8). Sur les données non bruitées (tableau 3.9), on remarque que cette erreur absolue moyenne ne représente que 5 % de l'amplitude maximale de la température relevées sur les données « Transistor » ; ceci indique une bonne adéquation des prédictions aux mesures réelles.

Méthode	Paramètres	erreur de validation	erreur quadratique moyenne en test	erreur absolue moyenne en test (°C)
Linéaire FIR	$n_u = 15$	192.833	186.104	10.261
Linéaire ARX	$n_u = 10$ et $n_y = 1$	192.989	187.470	10.301
Linéaire OE	$n_u = 4$ et $n_{\hat{y}} = 1$	191.610	185.831	10.249
NNOE	$n_u = 4$, $n_{\hat{y}} = 1$ et 1 neurone	191.878	185.740	10.235
LSSVM	$n_u = 4$, $n_y = 1$, $\sigma = 46.42$ et $C = 4.64$	210.969	203.765	10.820
SVR	$n_u = 4$, $n_y = 1$, $\sigma = 46.42$ et $C = 46.42$	211.199	203.292	10.800
RLSSVM	$n_u = 4$, $n_{\hat{y}} = 1$ et $\sigma = 46.42$	-	241.580	19.035
RB		235.907	209.803	10.934

TABLE 3.8 : Récapitulatif des résultats fournis par les différentes méthodes statistiques sur les données « Transistor » bruitées.

Méthode	Paramètres	erreur de validation	erreur quadratique moyenne en test	erreur absolue moyenne en test (°C)
Linéaire FIR	$n_u = 15$	192.833	80.576	4.474
Linéaire ARX	$n_u = 10$ et $n_y = 1$	192.989	82.453	4.784
Linéaire OE	$n_u = 4$ et $n_{\hat{y}} = 1$	191.610	80.184	4.382
NNOE	$n_u = 4$, $n_{\hat{y}} = 1$ et 1 neurone	191.878	80.152	4.326
LSSVM	$n_u = 4$, $n_y = 1$, $\sigma = 46.42$ et $C = 4.64$	210.969	99.324	6.464
SVR	$n_u = 4$, $n_y = 1$, $\sigma = 46.42$ et $C = 46.42$	211.199	98.395	6.384
RLSSVM	$n_u = 4$, $n_{\hat{y}} = 1$ et $\sigma = 46.42$	-	138.305	16.926
RB		235.907	104.580	6.594

TABLE 3.9 : Récapitulatif des résultats fournis par les différentes méthodes statistiques sur les données « Transistor » sans bruit.

Pour les données « Maquette », les résultats sont récapitulés dans la table 3.10. L'utilisation de modèles non-linéaires est plus justifiée sur ce jeu de données. Tout d'abord, l'écart entre les réseaux neurones et les modèles linéaires est plus important. De plus, le modèle SVR présente des performances plus proches de celles du modèle ARX en comparaison aux données « Transistor ». En moyenne, l'erreur absolue sur les données test est faible, de l'ordre 3 à 4 % des maxima de température constatés sur les données de « Maquette ». Toutefois, cette remarque mérite quelques commentaires. En effet, quand on compare la prédicition des modèles avec les mesures réelles, notamment sur la figure 3.12(b) qui décrit les sorties des modèles OE (linéaire et réseaux de neurones), on peut constater que ces modèles décrivent assez bien les régimes transitoires. En revanche, ces modèles peinent à approximer correctement les régimes permanents. De la même manière, l'influence du système de refroidissement (vitesse du ventilateur) est faiblement prise en compte par les modèles. Quelques pistes peuvent être esquissées pour résoudre ces problèmes : l'utilisation de scénarii d'entrées plus variées et plus complexes en durée et amplitude (ce qui n'est pas possible dans la configuration actuelle de la maquette) tout en réduisant la fréquence d'échantillonnage pourrait permettre de générer des bases de données plus riches (en termes d'excitation persitante des entrées) et plus exploitables dans le cadre d'un apprentissage statistique. Une autre solution consisterait à tester sur les données actuelles, des approches de modélisation dynamique comme les modèles de Hammerstein qui consistent à faire passer les entrées du système à travers un bloc linéaire, représentatif d'un gain statique non-linéaire, suivi par un système dynamique linéaire comme un ARX ou un OE. Ce faisant, l'on peut espérer améliorer la qualité de l'approximation des régimes permanents. Dans nos expériences sur les données« Maquette », nous avons choisi de considérer le même ordre n_u pour les deux entrées (puissance injectée et vitesse de l'air) afin de réduire la complexité du processus de sélection de modèles. Il est envisageable de s'affranchir de cette contrainte et considérer des ordres différents afin de mieux intégrer l'influence du système de refroidissement.

Méthode	Paramètres	erreur de validation	erreur quadratique moyenne en test	erreur absolue moyenne en test (°C)
Linéaire FIR	$n_u = 15$	4.908	3.866	1.620
Linéaire ARX	$n_u = 4$ et $n_y = 1$	-	3.955	1.702
Linéaire OE	$n_u = 4$ et $n_{\hat{y}} = 1$	4.032	3.403	1.513
NNOE	$n_u = 4$, $n_{\hat{y}} = 1$ et 2 neurones	3.229	2.535	1.402
LSSVM	$n_u = 4$, $n_y = 1$, $\sigma = 359.38$ et $C = 100$	4.053	6.501	2.185
SVR	$n_u = 4$, $n_y = 1$, $\sigma = 46.42$ et $C = 0.46$	4.816	4.575	1.628
RLSSVM	$n_u = 4$, $n_{\hat{y}} = 1$ et $\sigma = 359.38$	-	153.280	12.284
RB		13.390	7.536	2.057

TABLE 3.10 : Récapitulatif des résultats fournis par les différentes méthodes statistiques sur les données « Maquette ».

L'utilisation de modèles linéaires est donc satisfaisante sur les 2 jeux de données. Un réseau de neurones peut se justifier sur les données « Maquette » même si ce résultat mérite d'être confirmé sur des jeux de données plus riches où l'identification du modèle serait plus précise. Les autres méthodes non-linéaires utilisées ne semblent pas être adaptées à nos données. Une voie pour l'identification de modèles dynamiques par des méthodes à noyaux peut être l'utilisation de méthodes de sous-espace qui seront développées dans le chapitre 4. La flexibilité dans l'architecture des réseaux bayésiens ne semble pas être utile dans notre cas et cette méthode est de surcroit trop proche des modèles linéaires (ou de sous-espace dans le cas de plusieurs états cachés) pour espérer une réelle amélioration par rapport à ces méthodes déjà bien établies.

En conclusion, afin de confirmer ces résultats, nous avons choisi de réutiliser ces 2 méthodes (modèle linéaire et réseaux de neurones) en faisant également varier les données de test. Le protocole expérimental reste le même, mais la partie des données (dans le cas des données « Transistor ») ou la base (dans le cas des données « Maquette ») servant pour le test ont été changées plusieurs fois. Pour les données « Transistor », on a considéré différents découpages de la base de données initiales afin de générer 9 jeux de base Apprentissage/Validation et Test. Pour les données « Maquette », chaque base a servi alternativement de base de Test (donc 5 jeux de données Appren-

tissage/Validation différents). Les résultats obtenus en erreur quadratique moyenne de test sont présentés dans le tableau 3.11.

Méthode	Données « Transistor »	Données « Maquette »
Modèles linéaires OE	Paramètres : $n_u = 4.1 \pm 2.6$ et $n_{\hat{y}} = 1 \pm 0$ Erreur quadratique moyenne = 197.8 ± 12.6	Paramètres : $n_u = 8.6 \pm 2$ et $n_{\hat{y}} = 1 \pm 0$ Erreur quadratique moyenne = 4 ± 2.5
Réseaux de neurones NNOE	Paramètres : $n_u = 4$ et $n_{\hat{y}} = 1$ 2 ± 0 neurone(s) Erreur quadratique moyenne = 199.5 ± 14.7	Paramètres : $n_u = 9$ et $n_{\hat{y}} = 1$ 1.2 ± 0.4 neurone(s) Erreur quadratique moyenne = 3.8 ± 2.7

TABLE 3.11 : Récapitulatif des résultats fournis par les modèles linéaires et les réseaux de neurones en faisant varier les données de test.

Le premier enseignement de ce tableau est que le vecteur de régression ne retient à chaque fois qu'un seul retard sur la sortie prédite. Les modèles linéaires d'ordre supérieur ont en effet eu des résultats trop aléatoires pour être retenus. Les variations du nombre de retards sur l'entrée sont moins surprenantes, les résultats pour les 2 jeux de données sont très similaires pour les modèles au-delà de 3 retards sur l'entrée. Pour les réseaux de neurones, le vecteur de régression a été pris comme le meilleur obtenu pour les modèles linéaires. On observe toujours une très grande variabilité des résultats pour les cas où plusieurs neurones sont présents en couche cachée. Les réseaux obtenant les meilleurs résultats ont toujours 1 ou 2 neurones en couche cachée, ce qui semble confirmer la faible non-linéarité des données.

4

Identification de représentations d'état stables

4.1 Introduction

Une autre façon d'appréhender la modélisation des systèmes dynamiques est la représentation d'état. Cette représentation exprime la relation entre les entrées d'un système et ses sorties par une équation aux différences (pour les modèles à temps continu, par une équation différentielle) de premier ordre en utilisant une variable auxiliaire dénommée variable d'état (notons que cette idée est également sous-jacente des modèles de Markov dynamiques exposés dans la section 3.7). Pour un système linéaire à temps invariant, à entrées multiples $\mathbf{u}$ et sorties multiples $\mathbf{y}$, le modèle d'état sous sa forme discrète s'écrit

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k + \mathbf{v}_k \tag{4.1a}$$

$$\mathbf{y}_k = \mathbf{C}\mathbf{x}_k + \mathbf{D}\mathbf{u}_k + \mathbf{e}_k \tag{4.1b}$$

où $\mathbf{x}$ représente le vecteur d'état. Ce modèle suppose l'existence de bruits $\mathbf{v}$ affectant la relation d'état (4.1a) et de bruits de mesures $\mathbf{e}$ qui corrompent la relation de sortie (4.1b). La représentation d'état (4.1) présente l'avantage d'englober les structures de modèle linéaire (ARX, ARMAX, OE, ...) évoqués dans le tableau 3.3 et des détails sur l'équivalence de (4.1) avec les modèles pré-cités peuvent être trouvés dans [Lju02].

L'objectif des différentes techniques d'identification consiste à déterminer les matrices $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ et $\mathbf{D}$ du modèle à partir de mesures des entrées et des sorties du système dynamique [OM96, Ver94, Vib02, CM05, Lar90]. Ces techniques relèvent globalement de trois familles. La première famille procède par estimation des états $\mathbf{x}_k$, qui sont bien sûr inconnus, de laquelle on déduit les paramètres du modèle par une optimisation de type moindres carrés. La deuxième famille se base sur l'estimation de la matrice d'observabilité étendue (cette notion sera précisée ultérieurement dans ce chapitre) de laquelle on déduit $\mathbf{A}$ et $\mathbf{C}$. Les paramètres restants sont estimés à partir d'une matrice de Toepliz inférieure du système. La dernière famille repose sur l'estimation des paramètres de Markov du système desquels on extrait ensuite les paramètres recherchés. Ces méthodes d'identification font essentiellement appel à des outils d'algèbre linéaire (principalement des méthodes de projection des matrices formées à partir des entrées et sorties du système dans des espaces appropriés) qui ne tiennent pas compte de la nécessité que les paramètres identifiés doivent correspondre à une modélisation. Ce problème se pose avec d'autant plus d'acuité que le système considéré est à la limite de la stabilité. Il en résulte alors un modèle instable où à la limite de la stabilité. Ceci engendre une dérive de sa sortie lorsque le modèle est utilisé en erreur de sortie (OE).

L'objectif de ce chapitre est de proposer une approche permettant d'attaquer ce problème. Notre approche consiste à poser le problème d'identification sous contraintes de stabilité c'est-à-dire des contraintes liées au rayon spectral de la matrice $\mathbf{A}$. Le problème d'optimisation induit étant non-convexe, la plupart des auteurs proposent de substituer une ou plusieurs contraintes convexes approchant l'espace des solutions admissibles. Dans ce chapitre, nous prenons l'option de résoudre directement le problème d'optimisation non-convexe sous-jacent en nous basant

sur des approches dites de gradient échantillonné [BLO05] permettant de calculer le gradient d'un critère dépendant du rayon spectral.

La première partie de ce chapitre décrit les approches usuelles d'identification de modèles d'état. Après l'explicitation du problème de stabilité des modèles (4.1), nous ferons un tour d'horizon des techniques destinées à circonscrire le problème et les nouvelles approches que nous proposons. Finalement, le chapitre illustrera nos approches, en comparaison avec des méthodes existantes, sur un exemple tiré de la littérature.

4.2 Méthodes d'identification de type sous-espaces

Cette section est dédiée à un bref panorama des techniques existantes d'identification des systèmes dynamiques décrits par (4.1). Dans ces techniques, la démarche d'identification ne s'attache pas à déterminer absolument le quadruplet $(\mathbf{A},\mathbf{B},\mathbf{C},\mathbf{D})$ mais une réalisation équivalente $(\tilde{\mathbf{A}},\tilde{\mathbf{B}},\tilde{\mathbf{C}},\tilde{\mathbf{D}})$ avec les définitions suivantes $\tilde{\mathbf{A}} = \mathbf{T}^{-1}\mathbf{A}\mathbf{T}$, $\tilde{\mathbf{B}} = \mathbf{T}^{-1}\mathbf{B}$, $\tilde{\mathbf{C}} = \mathbf{C}\mathbf{T}$ et $\tilde{\mathbf{D}} = \mathbf{D}$ où $\mathbf{T}$ est une matrice inversible. Ce principe repose sur le fait qu'un système admet une infinité de réalisations d'état. En effet étant donné une réalisation liée au vecteur d'état $\mathbf{x}$, on peut passer à une réalisation équivalente basée sur les états $\tilde{\mathbf{x}} = \mathbf{T}^{-1}\mathbf{x}$ possédant les mêmes propriétés dynamiques que la réalisation initiale.

La démarche d'identification peut être schématisée par le diagramme de la figure 4.1. Elle fait intervenir un certain nombre de matrices dont la définition est nécessaire pour appréhender la quintessence de la procédure d'identification.

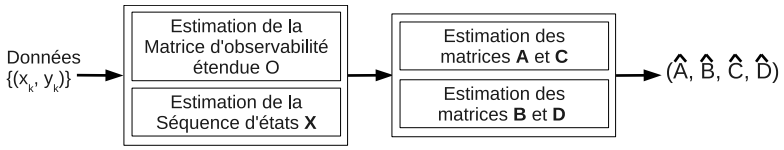


FIGURE 4.1 : Schéma résumant les procédures classiques d'identification par les méthodes de sous-espaces.

4.2.1 Définitions et notations

Considérons le système d'ordre n caractérisé par l'équation (4.1) où le vecteur d'état $\mathbf{x}_k \in \mathbb{R}^n$. De façon générique, considérons que les entrées du système sont multidimensionnelles i.e. $\mathbf{u}_k \in \mathbb{R}^m$ ainsi que ses sorties $\mathbf{y}_k \in \mathbb{R}^p$. Les matrices du modèle d'état ont évidemment les dimensions compatibles avec ces définitions. On définit alors différentes matrices présentées ci-dessous.

Matrice d'observabilité étendue Pour tout entier $r > n$, la matrice d'observabilité du système s'écrit

$$\mathbf{O}_r = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \vdots \\ \mathbf{CA}^{r-1} \end{bmatrix}. \tag{4.2}$$

Si l'ordre du système est n et le système est observable, cette matrice est de plein rang colonne et $\text{rank}(\mathbf{O}_r) = n$.

Matrice de Toeplitz La matrice de Toeplitz inférieure du système est définie par :

$$\mathbf{H}_r = \begin{bmatrix} \mathbf{D} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{CB} & \mathbf{D} & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{CA}^{r-2}\mathbf{B} & \mathbf{CA}^{r-3}\mathbf{B} & \cdots & \mathbf{CB} & \mathbf{D} \end{bmatrix}. \tag{4.3}$$

Matrices de Hankel Les méthodes d'identification par les sous-espaces font une utilisation intensive des matrices de Hankel. Ansi les matrices de Hankel d'entrée et de sorties sont définies par

$$\mathbf{U}_{t/r} = \begin{bmatrix} \mathbf{u}_{t+1} & \cdots & \mathbf{u}_{t+j} \\ \mathbf{u}_{t+2} & \cdots & \mathbf{u}_{t+j+1} \\ \cdots & \cdots & \cdots \\ \mathbf{u}_{t+r} & \cdots & \mathbf{u}_{t+r+j-1} \end{bmatrix}, \qquad \mathbf{Y}_{t/r} = \begin{bmatrix} \mathbf{y}_{t+1} & \cdots & \mathbf{y}_{t+j} \\ \mathbf{y}_{t+2} & \cdots & \mathbf{y}_{t+j+1} \\ \cdots & \cdots & \cdots \\ \mathbf{y}_{t+r} & \cdots & \mathbf{y}_{t+r+j-1} \end{bmatrix} \tag{4.4}$$

pour les entiers r et j tels que $r < j$. Pour $t = 0$, les matrices obtenues $\mathbf{U}_{0/r}$ et $\mathbf{Y}_{0/r}$ seront notées respectivement $\mathbf{U}_0$ et $\mathbf{Y}_0$ pour plus de simplicité. Elles sont souvent dénommées matrices des entrées et sorties passées. On en déduit la matrice de Hankel des données passées comme

$$\mathbf{Z}_0 = \begin{bmatrix} \mathbf{U}_0 \\ \mathbf{Y}_0 \end{bmatrix}. \tag{4.5}$$

En toute similitude, on appelle matrices de Hankel des données futures les matrices obtenues en considérant $t = r$, c'est-à-dire $\mathbf{U}_{r/r}$ et $\mathbf{Y}_{r/r}$, et on les notera simplement par $\mathbf{U}_r$ et $\mathbf{Y}_r$.

Séquence d'états En dehors de ces matrices, on définit la matrice des séquences d'état comme

$$\mathbf{X}_t = \begin{bmatrix} \mathbf{x}_{t+1} & \mathbf{x}_{t+2} & \cdots & \mathbf{x}_{t+j} \end{bmatrix}. \tag{4.6}$$

Dans la suite, on notera la matrice des séquences d'état passée et future respectivement par $\mathbf{X}_0$ et $\mathbf{X}_r$.

Matrices de projection Soit $\mathbf{N}$ une matrice rectangulaire à valeurs réelles. L'opérateur de projection sur l'espace des lignes de $\mathbf{N}$ est noté

$$\Pi_{\mathbf{N}} = \mathbf{N}^\top \left(\mathbf{N}\mathbf{N}^\top\right)^\dagger \mathbf{N}$$

où le terme $\mathbf{Q}^\dagger$ représente la pseudo-inverse de Moore-Penrose de la matrice $\mathbf{Q}$. Soit une matrice $\mathbf{M}$ quelconque de dimensions appropriées ; sa projection via cet opérateur est donnée par $\mathbf{M}\Pi_{\mathbf{N}}$.

L'opérateur de projection sur l'espace orthogonal à l'espace des lignes de la matrice $\mathbf{N}$ est alors défini par

$$\Pi_{\mathbf{N}}^\perp = \mathbf{I} - \Pi_{\mathbf{N}},$$

et on vérifie aisément que $\mathbf{N}\Pi_{\mathbf{N}}^\perp = \mathbf{0}$.

4.2.2 Estimation de la matrice d'observabilité étendue et de la séquence d'états

La détermination de la matrice d'observabilité étendue découle de l'utilisation des équations du modèle d'état (4.1) à partir desquelles on peut écrire

$$\begin{aligned} \mathbf{y}_{t+1+k} &= \mathbf{C}\mathbf{x}_{t+1+k} + \mathbf{D}\mathbf{u}_{t+1+k} + \mathbf{e}_{t+1+k} && (4.7) \\ &= \mathbf{C}\mathbf{A}^k\mathbf{x}_{t+1} + \mathbf{C}\mathbf{A}^{k-1}\mathbf{B}\mathbf{u}_{t+1} + \mathbf{C}\mathbf{A}^{k-2}\mathbf{B}\mathbf{u}_{t+2} + \cdots + \mathbf{C}\mathbf{B}\mathbf{u}_{t+k} + \mathbf{D}\mathbf{u}_{t+1+k} && (4.8) \\ &\quad + \underbrace{\mathbf{C}\mathbf{A}^{k-1}\mathbf{v}_{t+1} + \mathbf{C}\mathbf{A}^{k-2}\mathbf{v}_{t+2} + \cdots + \mathbf{C}\mathbf{v}_{t+k} + \mathbf{e}_{t+1+k}}_{\xi_{t+1}} \end{aligned}$$

En compilant successivement cette relation, il est aisé d'établir la relation matricielle usuelle des méthodes d'identification par les sous-espaces

$$\mathbf{Y}_r = \mathbf{O}_r\mathbf{X}_r + \mathbf{H}_r\mathbf{U}_r + \Xi_r \tag{4.9}$$

où la matrice Ξ_r regroupant tous les termes relatifs aux bruits est définie conformément à (4.4) à partir de l'expression de ξ_k dans l'équation (4.7). Il est à noter que Ξ_r est une forme linéaire des matrices de Hankel futures construites à partir des mesures.

L'estimation de la matrice d'observabilité étendue repose sur la projection orthogonale de cette équation sur l'espace des entrées de façon à supprimer l'influence des entrées. En effet, en multipliant à droite l'équation précédente par $\Pi^{\perp}_{\mathbf{U}_r}$, on aboutit à

$$\mathbf{Y}_r\Pi^{\perp}_{\mathbf{U}_r} = \mathbf{O}_r\mathbf{X}_r\Pi^{\perp}_{\mathbf{U}_r} + \Xi_r\Pi^{\perp}_{\mathbf{U}_r}.$$

En supposant les bruits de mesure et d'état blancs, stationnaires et décorrelés des entrées (la décorrélation des bruits par rapport aux entrées est dans la pratique vraie si les données d'apprentissage ont été récoltées en boucle ouverte), on montre que l'on a

$$\mathbb{E}(\Xi_r\Pi^{\perp}_{\mathbf{U}_r}) = \mathbf{0}$$

c'est-à-dire qu'asymptotiquement, l'estimation de $\mathbf{O}_r\mathbf{X}_r$ est non biaisée. En pratique, pour améliorer cette estimation, on considère deux matrices de pondération $\mathbf{W}_1$ et $\mathbf{W}_2$ vérifiant les conditions suivantes [OM96, Pek04, Lju02] :

- $\mathbf{W}_1$ est inversible,
- $\text{rang}(\mathbf{X}_r\Pi^{\perp}_{\mathbf{U}_r}\mathbf{W}_2) = \text{rang}(\mathbf{X}_r) = n$,
- les matrices de pondération sont telles qu'elles sont décorrelées avec la matrice de bruit future Ξ_r.

On peut alors écrire la relation

$$\begin{aligned}\mathbf{W}_1\mathbf{Y}_r\Pi^{\perp}_{\mathbf{U}_r}\mathbf{W}_2 &= \mathbf{W}_1\mathbf{O}_r\mathbf{X}_r\Pi^{\perp}_{\mathbf{U}_r}\mathbf{W}_2 + \mathbf{W}_1\Xi_r\Pi^{\perp}_{\mathbf{U}_r}\mathbf{W}_2 \\ &= \mathbf{W}_1\mathbf{O}_r\tilde{\mathbf{T}} + \tilde{\Xi}_r\end{aligned} \tag{4.10}$$

qui permet d'obtenir une estimation bruitée d'une réalisation de la matrice d'observabilité (c'est-à-dire une estimation de $\mathbf{O}_r$ à la transformation $\tilde{\mathbf{T}}$ près). L'ordre du système correspond à $n = \text{rang}(\mathbf{W}_1\mathbf{Y}_r\Pi^{\perp}_{\mathbf{U}_r}\mathbf{W}_2)$. En pratique l'ordre du système est choisi à partir de la décomposition en valeurs singulières (voir annexe B)

$$\mathbf{W}_1\mathbf{Y}_r\Pi^{\perp}_{\mathbf{U}_r}\mathbf{W}_2 = \bar{\mathbf{U}}\bar{S}\bar{\mathbf{V}}^{\top} \approx \bar{\mathbf{U}}_1\bar{\mathbf{S}}_1\bar{\mathbf{V}}_1^{\top}$$

avec $\bar{\mathbf{S}}_1$ la matrice des valeurs singulières les plus significatives. L'estimation de la matrice d'observabilité étendue est obtenue via

$$\hat{\mathbf{O}}_r = \mathbf{W}_1^{-1}\bar{\mathbf{U}}_1\bar{\mathbf{S}}_1^{1/2} \tag{4.11}$$

et une estimation de la séquence des états correspondante est

$$\hat{\mathbf{X}}_r = \mathbf{X}_r\Pi^{\perp}_{\mathbf{U}_r}\mathbf{W}_2 = \bar{\mathbf{S}}_1^{1/2}\bar{\mathbf{V}}_1^{\top}. \tag{4.12}$$

Les principales méthodes des sous-espaces diffèrent par les choix des matrices de pondération [OM95] comme l'expose le tableau 4.1.

4.2.3 Estimation d'une réalisation du système

La détermination d'une réalisation des matrices du système peut se faire en exploitant soit la connaissance de la matrice d'observabilité étendue pour estimer $\mathbf{A}$ et $\mathbf{C}$ puis en déduire celle de $\mathbf{B}$ et $\mathbf{D}$ (algorithme MOESP), soit l'estimation de la séquence d'état pour déduire les matrices du système par moindres carrés (CVA et N4SID). Nous explicitons brièvement ces deux approches ci-dessous.

Technique	$\mathbf{W}_1$	$\mathbf{W}_2$
MOESP [Ver94]	$\mathbf{I}$	$\mathbf{Z}_0^\top \left(\mathbf{Z}_0 \Pi_{\mathbf{U}_r}^\perp \mathbf{Z}_0^\top\right)^{-1} \mathbf{Z}_0 \Pi_{\mathbf{U}_r}^\perp$
N4SID [OM94]	$\mathbf{I}$	$\mathbf{Z}_0^\top \left(\mathbf{Z}_0 \Pi_{\mathbf{U}_r}^\perp \mathbf{Z}_0^\top\right)^{-1} \mathbf{Z}_0$
CVA [Lar90]	$\left(\mathbf{Y}_r \Pi_{\mathbf{U}_r}^\perp \mathbf{Y}_r^\top\right)^{-1/2}$	$\mathbf{Z}_0^\top \left(\mathbf{Z}_0 \Pi_{\mathbf{U}_r}^\perp \mathbf{Z}_0^\top\right)^{-1} \mathbf{Z}_0 \Pi_{\mathbf{U}_r}^\perp$

TABLE 4.1 : Choix des matrices de pondération dans les principales méthodes d'identification par les sous-espaces. L'expression des matrices suppose implicitement que $\mathbf{Z}_0 \Pi_{\mathbf{U}_r}^\perp$ est de plein rang lignes.

Utilisation de la matrice d'observabilité étendue En se basant sur la définition (4.2) de la matrice d'observanilité étendue, et connaissant $\hat{\mathbf{O}}_r$, on obtient de façon directe (en utilisant une notation de type Matlab)

$$\hat{\mathbf{C}} = \hat{\mathbf{O}}_r(1:p, 1:n).$$

L'estimation de la matrice d'état $\mathbf{A}$ repose sur la propriété de $\mathbf{A}$-invariance de la matrice d'observabilité étendue. En effet, il est aisé de constater que $\mathbf{O}_r(p+1:pr, 1:n) = \mathbf{O}_r(1:p(r-1), 1:n)\mathbf{A}$. On en déduit alors l'estimation au sens des moindres carrés

$$\hat{\mathbf{A}} = \hat{\mathbf{O}}_r^\dagger(1:p(r-1), 1:n)\, \hat{\mathbf{O}}_r(p+1:pr, 1:n). \tag{4.13}$$

Le calcul des autres matrices du système peut se faire selon deux stratégies :

1. du modèle d'état (4.1), on déduit qu'une prédiction déterministe de la sortie du système s'écrit $\hat{\mathbf{y}}_k = \hat{H}(q, \mathbf{B}, \mathbf{D})\mathbf{u}_k$ où $\hat{H}(q, \mathbf{B}, \mathbf{D}) = \hat{\mathbf{C}}\left(q\mathbf{I} - \hat{\mathbf{A}}\right)\mathbf{B} + \mathbf{D}$ représente la "fonction de transfert" du système et est linéaire par rapport à $\mathbf{B}$ et $\mathbf{D}$. L'estimation s'obtient par minimisation du critère quadratique $J(\mathbf{B}, \mathbf{D}) = \sum_{k=1}^{N} (\mathbf{y}_k - \hat{\mathbf{y}}_k)^2$,
2. la deuxième approche est basée sur l'estimation de la matrice de Toeplitz $\mathbf{H}_r$ (équation 4.3) connaissant $\mathbf{A}$ et $\mathbf{C}$. En effet, à partir de l'expression (4.9), on peut écrire

   $$\Pi_{\mathbf{O}_r}^\perp \mathbf{Y}_r \mathbf{U}_r^\dagger = \Pi_{\mathbf{O}_r}^\perp \mathbf{H}_r + \Pi_{\mathbf{O}_r}^\perp \Xi_r \mathbf{U}_r^\dagger.$$

   De part la structure de $\mathbf{H}_r$, on constate que cette dernière relation est linéaire en $\mathbf{B}$ et $\mathbf{D}$. Par conséquent les matrices manquantes du système sont calculées via un problème des moindres carrés.

Utilisation de la séquence d'états Si une estimation des états est disponible, on peut écrire à partir de (4.1) et de la définition des matrices de séquence d'état (4.6) et de Hankel (4.4)

$$\left[\frac{\hat{\mathbf{X}}_{t+1}}{\mathbf{Y}_{1|t}}\right] = \Theta \left[\frac{\hat{\mathbf{X}}_t}{\mathbf{U}_{1|t}}\right] + \left[\frac{\rho_{\mathbf{v}}}{\rho_{\mathbf{e}}}\right]$$

où le dernier terme du membre de droite est relatif aux bruits. Les matrices inconnues du système sont rangées dans

$$\Theta = \left[\begin{array}{c|c} \mathbf{A} & \mathbf{B} \\ \hline \mathbf{C} & \mathbf{D} \end{array}\right]$$

et sont déterminées à travers la solution du problème des moindres carrés

$$\min_{\Theta} \left\| \left[\frac{\hat{\mathbf{X}}_{t+1}}{\mathbf{Y}_{1|t}}\right] - \Theta \left[\frac{\hat{\mathbf{X}}_t}{\mathbf{U}_{1|t}}\right] \right\|_F^2 \tag{4.14}$$

où $\|\cdot\|_F$ représente la norme de Frobenius.

Toute la problématique réside dans l'estimation de la séquence d'état. Van Overschee et De Moor [OM96] montrent dans l'approche N4SID que pour les choix de matrices de pondération tels que définis dans le tableau 4.1, l'expression (4.10) est équivalente à $\mathbf{G}_r = \mathbf{W}_1 \mathbf{Y}_r \Pi_{\mathbf{U}_r}^\perp \mathbf{W}_2 = \mathbf{O}_r \hat{\mathbf{X}}_r$ avec $\hat{\mathbf{X}}_r$ une estimation de la séquence d'état fournie par un banc de filtres de Kalman avec des conditions initiales particulières. Disposant de l'estimation (4.11) de la matrice d'observabilité étendue, la séquence d'état s'obtient par $\hat{\mathbf{X}}_r = \mathbf{O}_r^\dagger \mathbf{G}_r$.

Pour déterminer la séquence d'états $\mathbf{X}_{r+1}$, on conduit un raisonnement similaire et on montre que [OM96]

$$\mathbf{G}_{r+1} = \mathbf{W}_1 \mathbf{Y}_{r+1/r} \Pi^{\perp}_{\mathbf{U}_{r+1/r}} \bar{\mathbf{W}}_2 = \mathbf{O}_{r-1} \hat{\mathbf{X}}_{r+1} \tag{4.15}$$

où la matrice de pondération $\bar{\mathbf{W}}_2 = \mathbf{Z}_1^{\top} \left(\mathbf{Z}_1 \Pi^{\perp}_{\mathbf{U}_{r+1/r}} \mathbf{Z}_1^{\top} \right)^{-1} \mathbf{Z}_1$ (selon le tableau 4.1 pour N4SID) dépend de la nouvelle matrice des données passées

$$\mathbf{Z}_1 = \begin{bmatrix} \mathbf{U}_{0/r+1} \\ \mathbf{Y}_{0/r+1} \end{bmatrix}.$$

Une estimation de la matrice d'observabilité dans (4.15) s'obtient simplement à partir de $\hat{\mathbf{O}}_r$ par

$$\hat{\mathbf{O}}_{r-1} = \hat{\mathbf{O}}_r(1 : p(r-1), 1 : n).$$

On en déduit la deuxième séquence d'état

$$\hat{\mathbf{X}}_{r+1} = \mathbf{O}^{\dagger}_{r-1} \mathbf{G}_{r+1}$$

dont la connaissance permet de mettre en oeuvre l'estimation des moindres carrés (4.14).

Cette procédure de détermination des séquences d'état est adaptable à la méthode CVA moyennant quelques légères modifications dans les matrices de pondération. Le lecteur pourra consulter l'ouvrage de Van Overschee et De Moor [OM96] pour de plus amples détails.

Remarque

D'autres méthodes d'estimation par les méthodes des sous-espaces moins usitées existent et se basent sur l'estimation consistante des paramètres de Markov (éléments de la matrice de Toeplitz 4.3) desquels est déduite une réalisation du système. Des exemples de telles approches sont exposés par exemple dans [CM05] et [Pek04]. Elles reposent comme les méthodes exposées précédemment sur les outils de projection de l'algèbre linéaire et sont susceptibles comme les autres méthodes des sous-espaces de produire des modèles instable du système. Cette remarque nous conduit à examiner la notion de stabilité des systèmes linéaires à temps invariant.

4.3 Stabilité des systèmes linéaires à temps invariant

Dans un cadre général, la notion de stabilité d'un système décrit par les équations

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k) \tag{4.16}$$

$$\mathbf{y}_k = g(\mathbf{x}_k, \mathbf{u}_k) \tag{4.17}$$

où f et g sont des fonctions des états $\mathbf{x} \in \mathscr{X}$ et des entrées $\mathbf{u}$, est considérée par rapport aux points d'équilibre $(\mathbf{x}_e, \mathbf{u}_e)$ du système vérifiant la relation $\mathbf{x}_e = f(\mathbf{x}_e, \mathbf{u}_e)$. Pour simplifier la présentation, nous considérerons par la suite que $\mathbf{u}_e = 0$. Dans ce cas, la caractérisation de la stabilité du point d'équilibre $\mathbf{x}_e$ peut être résumée par les éléments de définition suivants.

Definiton 4.1 (Stabilité)**.** *Considérons la notation $\mathbf{x}_k(\mathbf{x}_0)$ qui désigne la trajectoire de l'état du système à l'instant $k > k_0$, partant de l'état initial $\mathbf{x}_{k_0} = \mathbf{x}_0$. Un point d'équilibre $\mathbf{x}_e$ est*

1. *simplement stable (au sens de Lyapunov) si $\forall \varepsilon > 0$, $\exists \eta(\varepsilon) > 0$ tel que pour tout état initial $\mathbf{x}_0$ vérifiant $\|\mathbf{x}_0 - \mathbf{x}_e\| < \eta$, $\|\mathbf{x}_{k+1}(\mathbf{x}_0) - \mathbf{x}_e\| < \varepsilon$*
2. *asymptotiquement stable s'il est simplement stable et $\lim_{k\to\infty} \|\mathbf{x}_{k+1}(\mathbf{x}_0) - \mathbf{x}_e\| = 0$,*
3. *globalement asymptotiquement stable si le système est asymptotiquement stable pour tout état initial appartenant au domaine admissible des états.*

4. *instable dans le cas contraire.*

Dans cette définition, la stabilité simple signifie simplement que pour tout point initial $\mathbf{x}_0 \in \Omega(\mathbf{x}_e) \subset \mathscr{X}$ avec $\Omega(\mathbf{x}_e)$ désignant un voisinage proche de l'état d'équilibre $\mathbf{x}_e$, la trajectoire $\mathbf{x}_k(\mathbf{x}_0)$ évoluera à proximité du point d'équilibre. La stabilité asymptotique assure en plus qu'en régime asymptotique, le système revient à l'état d'équilibre et elle se mue en stabilité globale asymptotique si cette propriété est vraie pour tout état initial $\mathbf{x}_0 \in \mathscr{X}$.

Une autre façon de caractériser la stabilité d'un système entièrement basée sur les entrées et sorties du système et ne faisant pas appel à l'état interne $\mathbf{x}$ est présentée ci-dessous.

Definiton 4.2 (Stabilité BIBO (Bounded Input, Bounded Output))**.** *Un système est BIBO stable si pour toute entrée bornée* $\|\mathbf{u}_k\| < \gamma$, $\gamma \geq 0$, *la sortie correspondante est bornée c'est-à-dire* $\|\mathbf{y}_k\| < \gamma'$ *avec* $\gamma' \geq 0$.

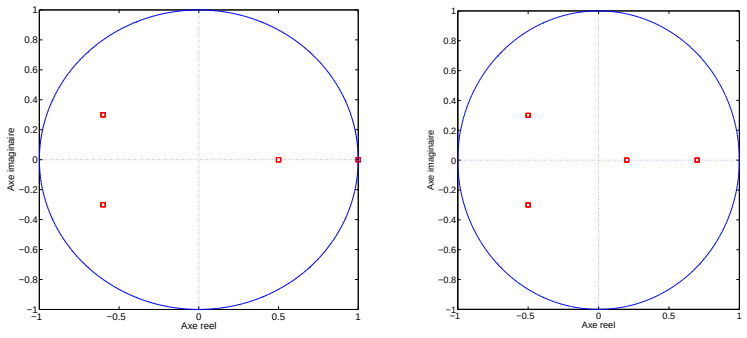
Dans le cas des systèmes linéaires comme ceux auxquels nous nous intéressons, le système admet en général un point d'équilibre unique et la notion de stabilité au sens de la définition 4.1 est une stabilité asymptotique globale, ce qui facilite l'étude. Nous exposons ci-après deux théorèmes spécifiant ces conditions de stabilité. Le premier théorème s'énonce de la façon qui suit.

Théorème 4.1 (Conditions de stabilité)**.** *Soit le système caractérisé par la relation d'état* $\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k$. *Soient* λ_j, $j = 1, \cdots, n'$, $n' \leq n$ *les valeurs propres de la matrice d'état* $\mathbf{A}$. *Ce système est*

1. *BIBO instable s'il existe au moins une valeur propre* λ_i *telle que* $|\lambda_i| > 1$,
2. *BIBO stable (et asymptotiquement stable) si toutes les valeurs propres vérifient* $|\lambda_j| < 1$, $\forall j = 1, \cdots, n'$,
3. *simplement stable (et BIBO non stable) si toutes les valeurs propres de* $\mathbf{A}$ *satisfont* $|\lambda_j| < 1$ *à l'exception d'une valeur propre* λ_i *telle que* $|\lambda_i| = 1$ *et est d'ordre de multiplicité 1.*

Les éléments de démonstration de ce théorème considèrent le système avec une entrée nulle. Par conséquent, le vecteur d'état suit une trajectoire décrite par l'équation $\mathbf{x}_k = \mathbf{A}^k \mathbf{x}_0$. En décomposant la matrice d'état sous sa forme de Jordan notée $\mathbf{J}(\lambda_1, \cdots, \lambda_{n'})$ (notons que $\mathbf{J}(\lambda_1, \cdots, \lambda_{n'})$ est diagonale si $\mathbf{A}$ est diagonalisable), on s'aperçoit que l'évolution de $\mathbf{x}_k$ ne dépend que de $\mathbf{J}^k(\lambda_1, \cdots, \lambda_{n'})$. De cette remarque, on déduit alors les résultats du théorème.

Une illustration de ce théorème est portée sur la figure 4.2.



(a) Stabilité simple (b) Stabilité asymptotique

FIGURE 4.2 : Illustration de la stabilité d'un système linéaire en fonction de la position dans le plan complexe des valeurs propres de la matrice d'état.

La deuxième méthode d'analyse de la stabilité repose sur la méthode de Lyapunov qui est une méthode générale d'analyse de la stabilité des systèmes qu'ils soient linéaires ou non. L'intuition derrière cette approche est la suivante : si l'énergie totale d'un système est dissipée de manière conitnue alors ce système finira par rejoindre un état d'équilibre. Cette idée est formalisée par le théorème suivant (que nous énonçons pour le point d'équilibre $\mathbf{x}_e = \mathbf{0}$ pour simplifier la présentation).

Théorème 4.2. *Soit $V_k = V(\mathbf{x}_k)$, une fonction candidate de Lyapunov et soit $\Delta V_k = V_{k+1} - V_k$ sa variation. Soit le système (4.16) de point d'équilibre $\mathbf{x}_e = \mathbf{0}$.*

1. *$\mathbf{x}_e$ est simplement stable si dans un voisinage de $\mathbf{x}_e$ (boule de rayon ρ centrée sur $\mathbf{x}_e$), il existe une fonction V_k définie positive [1] telle que ΔV_k soit semi-définie négative.*
2. *$\mathbf{x}_e$ est localement asymptotiquement stable si dans un voisinage de $\mathbf{x}_e$, il existe une fonction V_k définie positive telle que ΔV_k soit définie négative.*
3. *$\mathbf{x}_e$ est globalement asymptotiquement stable s'il existe une fonction V_k définie positive et non bornée en rayon (i.e. $\lim_{\|x_k\|\to\infty} V_k = +\infty$) telle que ΔV_k soit définie négative.*

Il est à remarquer que les conditions de stabilité de Lyapunov sont des conditions suffisantes. Dans le cas général, ce théorème pose le problème de définition de la fonction candidate de Lyapunov. Toutefois dans un cas linéaire comme (4.1), la mise en oeuvre du théorème est simplifiée. Pour ce faire, on choisit une fonction quadratique $V_k = \mathbf{x}_k^\top \mathbf{P}\mathbf{x}_k$ avec $\mathbf{P} \in \mathbb{R}^{n\times n}$. Sa variation s'écrit $\Delta V_k = \mathbf{x}_k^\top (\mathbf{A}^\top \mathbf{P}\mathbf{A} - \mathbf{P})\mathbf{x}_k$. Les conditions de stabilité s'énoncent alors comme suit.

Théorème 4.3 (Stabilité des systèmes linéaires par Lyapunov)**.** *Soit le système linéaire (4.1) et soit $V_k = \mathbf{x}_k^\top \mathbf{P}\mathbf{x}_k$, une fonction candidate de Lyapunov.*

1. *Le système est simplement stable ssi pour toute matrice $\mathbf{Q}$ symétrique, semi-définie positive ($\mathbf{Q} \succeq 0_n$), il existe une matrice $\mathbf{P}$ définie positive ($\mathbf{P} \succ 0_n$) vérifiant l'équation de Lyapunon*

$$\mathbf{A}^\top \mathbf{P}\mathbf{A} - \mathbf{P} + \mathbf{Q} = \mathbf{0} \tag{4.18}$$

2. *le système est asymptotiquement stable ssi pour toute matrice $\mathbf{Q}$ symétrique, définie positive ($\mathbf{Q} \succ 0_n$), il existe une matrice $\mathbf{P} \succ 0_n$ vérifiant l'équation (4.18).*

Remarque

Pour ce dernier théorème, on peut remarquer que pour $\mathbf{P} = \mathbf{I}_n$ s'il existe $\mathbf{Q} \succ 0_n$, on a $\mathbf{A}^\top \mathbf{A} \prec \mathbf{I}_n$ d'où on déduit que la norme spectrale $\|\mathbf{A}\|_2 < 1$. Ceci signifie que $\|\mathbf{x}_k\| \leq \|\mathbf{A}\|_2^k \|\mathbf{x}_0\|$ et donc que la convergence asymptotique de la séquence d'état vers 0 est garantie. En revanche le système peut être stable asymptotiquement sans qu'on ait nécessairement $\|\mathbf{A}\|_2 < 1$.

Notations

Pour la suite du chapitre, il est utile de définir plusieurs termes. Afin de mieux visualiser ces différentes notions, on présente une vue imagée en figure 4.3 . On suppose que l'on cherche une matrice $\mathbf{A}^*$, solution optimale d'un problème de moindres carrés sous contrainte de stabilité. $\hat{\mathbf{A}}$ est alors la solution du problème de moindres carrés sans contrainte. Soit $\lambda(\mathbf{A})$ une valeur propre de la matrice $\mathbf{A}$, $\lambda_{max}(\mathbf{A})$ désigne alors la plus grande valeur propre en module de la matrice $\mathbf{A}$. $\rho(\mathbf{A})$ est le module de cette valeur propre, également appelé rayon spectral de $\mathbf{A}$. On peut alors écrire : $\rho(\mathbf{A}) = |\lambda_{max}(\mathbf{A})|$. La contrainte de stabilité sur le système est alors donnée par l'inéquation $\rho(\mathbf{A}) < 1$. De même, $\sigma(\mathbf{A})$ désigne une valeur singulière de la matrice $\mathbf{A}$ et $\sigma_{max}(\mathbf{A})$ désigne la plus grande valeur singulière de la matrice $\mathbf{A}$. On peut remarquer ici que $\|\mathbf{A}\|_2 < 1$ est équivalent à $\sigma_{max}(\mathbf{A}) < 1$. On définit alors 2 régions de l'espace des matrices carrés de dimension n, S_λ l'ensemble des matrices $\mathbf{A}$ telles que $\rho(\mathbf{A}) \leq 1$, qui correspondra donc à l'ensemble des matrices accociées à la contrainte de stabilité choisie, et S_σ l'ensemble des matrices $\mathbf{A}$ telles que $\sigma_{max}(\mathbf{A}) \leq 1$. Concernant ces 2 régions, on peut noter que la région S_λ est non convexe. De plus, l'inégalité $\sigma_{max}(\mathbf{A}) \geq \rho(\mathbf{A})$ étant toujours vérifiée, la région S_σ est entièrement incluse dans S_λ.

1. La fonction $V(z)$ de $\mathbb{R}^n$ dans $\mathbb{R}^+$ est définie positive si $V(z) > 0$ pour tout $z \neq 0$ et $V(0) = 0$

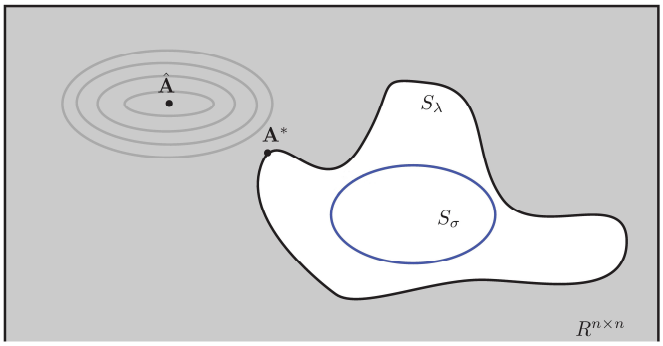


FIGURE 4.3 : Vue imagée des notations utilisées. En noir, la contrainte réelle de stabilité (l'espace des solutions admissibles est S_λ) et en bleu, la région désignée par S_σ. La matrice $\hat{\mathbf{A}}$ représente la solution des moindres carrés, entourée des isocontours du coût quadratique. $\mathbf{A}^*$ représente la solution optimale du problème du problème quadratique sous contrainte de stabilité.

4.4 Identification par méthodes des sous-espaces et problèmes de stabilité

L'application des procédures d'identification décrites dans la section 4.2 conduit généralement à des modèles stables lorsque l'on dispose d'un nombre suffisant de données et si le système dont sont issues les données est linéaire avec un ordre n raisonnable et stable. En revanche si l'ordre du système devient élevé ou si le système est faiblement stable (avec des pôles oscillatoires amorties proches du cercle unité) ou si le système est non-linéaire, le modèle identifié peut être instable [OM96, chapitre 4].

L'une des toutes premières approches garantissant la stabilité lorsque l'on travaille avec un nombre fini de données est due à Maciejowski [Mac95] qui propose une solution simple. Elle consiste à calculer une estimation de la matrice d'état sous la forme

$$\hat{\mathbf{A}} = \hat{\mathbf{O}}_r^{\dagger}\hat{\mathbf{O}}_0 \tag{4.19}$$

avec $\hat{\mathbf{O}}_0 = \begin{bmatrix} \hat{\mathbf{O}}_r(p+1:pr, 1:n) \\ \mathbf{0} \end{bmatrix}$.

Maciejowski a établi que cette légère modification portée à l'estimation classique (4.13) suffit à garantir que toute valeur propre $|\lambda(\hat{\mathbf{A}})| \leq 1$. Toutefois, Van Overschee et De Moor [OM96] recommandent d'utiliser avec précaution (4.19) car cette procédure a tendance à pousser les valeurs propres vers l'origine et à biaiser leur estimation. Le modèle résultant, bien que stable présente une dynamique différente de celle du système.

Pour la suite de ce chapitre, les matrices C et D de l'équation 4.14 n'étant pas concernées, il est utile de définir le coût J_1 tel que :

$$J_1(\mathbf{A},\mathbf{B}) = \left\| \hat{\mathbf{X}}_{t+1} - [\mathbf{A}\ \mathbf{B}] \begin{bmatrix} \mathbf{X}_t \\ \hline \mathbf{U}_{t|t} \end{bmatrix} \right\|_F^2 \tag{4.20}$$

Les matrices $\mathbf{A}$ et $\mathbf{B}$ sont alors déterminées par :

$$\min_{\mathbf{A},\mathbf{B}} J_1(\mathbf{A},\mathbf{B}) \tag{4.21}$$

4.5 Approches existantes

4.5.1 Contrainte convexe sur la valeur singulière de A (méthodes LB)

Lacy et Bernstein [LB02, LB03] ont présenté deux approches utilisant l'optimisation sous contrainte de stabilité pour résoudre ce problème. Ces méthodes se sont révélées plus performantes que celles existantes et procèdent par convexification de la contrainte de stabilité. L'espace de recherche est décrit à l'aide des inégalité de Lyapunov $\mathbf{P} - \mathbf{APA}^\top \succeq 0_n$, $\mathbf{P} \succeq 0_n$, qui sont respectées pour un système asymptotiquement stable ($\rho(\mathbf{A}) \leq 1$). Dans [LB02], Lacy et Bernstein choisissent un cas particulier $\mathbf{P} = \mathbf{I}_n$. Les contraintes s'écrivent alors sous la forme de l'inégalité matricielle $\mathbf{I}_n - \mathbf{AA}^\top \succeq 0_n$. Cette contrainte borne les valeurs propres de la matrice $\mathbf{AA}^\top$, et donc les valeurs singulière de la matrice $\mathbf{A}$, à 1 [2]. L'inégalité peut être mise sous la forme $\mathbf{S} - \mathbf{AA}^\top \succeq 0_n, I_n - S \succeq 0_n$. En utilisant le complément de Schur, cette contrainte est équivalente à :

$$\begin{bmatrix} \mathbf{S} & \mathbf{A} \\ \mathbf{A}^\top & \mathbf{I}_n \end{bmatrix} \succeq 0_{2n} \tag{4.22}$$

Le problème de moindres carrés (4.21) est réécrit pour intégrer cette contrainte. Comme la contrainte (4.22) est convexe, la solution peut être trouvée à l'aide d'outils de programmation quadratique sous contrainte de semi-positivité (toolbox Sedumi [Stu98]). Une vue imagée de cette contrainte permet de visualiser les différents ensembles (figure 4.4). Cependant, cette formulation implique une augmentation importante de la taille du problème à résoudre et la contrainte utilisée est trop restrictive par rapport à la contrainte de stabilité.

Pour surmonter ce problème, les mêmes auteurs ont adapté le principe d'optimisation sous contraintes sur un espace convexe défini par les inégalités [LB03] : $\mathbf{P} - \mathbf{APA}^\top \succeq \delta \mathbf{I}_n$, $\mathbf{P} \succeq \delta \mathbf{I}_n$. En utilisant le complément de Schur, on peut écrire :

$$\begin{bmatrix} \mathbf{P} - \delta \mathbf{I}_n & \mathbf{AP} \\ \mathbf{PA}^\top & \mathbf{P} \end{bmatrix} \succeq 0_{2n}, \tag{4.23}$$

Pour simplifier l'écriture du problème, on peut considérer $\mathbf{Q} = \mathbf{AP}$. Cependant, ce changement introduit une distorsion dans le problème des moindres carrés (4.21). Ainsi, même si cette méthode (nommée par la suite LB2) réduit la taille du problème d'optimisation et permet de trouver une solution à l'extérieur de l'ensemble $\sigma_{\max}(\mathbf{A}) \leq 1$ (où $\sigma_{\max}(A)$ correspond à la plus grande valeur singulière de $\mathbf{A}$) de l'espace des matrices carrés, l'augmentation de l'erreur de reconstruction est trop importante pour le rendre intéressant par rapport à la méthode précédente (nommé LB1).

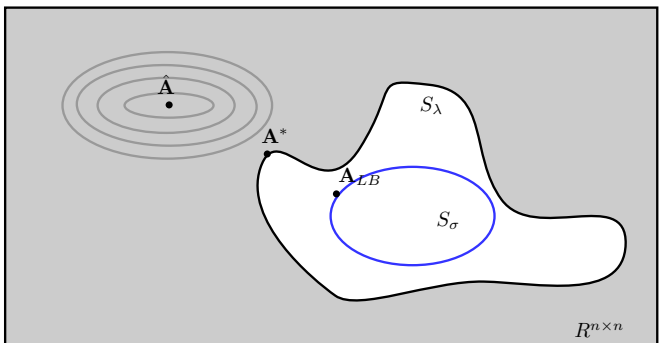


FIGURE 4.4 : Vue imagée des contraintes utilisées pour les algorithmes LB. En bleu, la contrainte utilisée dans LB1 (représentée par l'espace S_σ). $\mathbf{A}_{LB}$ représente la solution trouvée par l'algorithme LB1.

2. La contrainte $\mathbf{I}_n - \mathbf{AA}^\top \succeq 0_n$ implique que $\forall x \in \mathbb{R}$, $x\mathbf{AA}^\top x^\top$. En particulier, pour le vecteur propre $\mathbf{u}_1$ associé à la plus grande valeur propre λ_1 de $\mathbf{AA}^\top$, on obtient $\mathbf{u}_1\mathbf{AA}^\top\mathbf{u}_1^\top \leq 1 \Rightarrow \mathbf{u}_1\lambda_1\mathbf{u}_1^\top \leq 1 \Rightarrow \lambda_1 \leq 1$. On en déduit alors que $\rho(\mathbf{A}) \leq 1$ puisque $\forall \mathbf{A}$ carrée, $\rho(\mathbf{A}) \leq \|A\|_2$

4.5.2 Ajout itératif de contraintes linéaires (méthode CG)

Ces observations ont conduit Siddiqi et al. [SBG08] à essayer de trouver une méthode pour contourner ces inconvénients. Les deux principaux sont la taille du problème d'optimisation et un espace solution trop restrictif. Leur solution, nommée CG par la suite, consiste à remplacer l'espace convexe des matrices $\mathbf{A}$ telles que $\sigma_{\max}(\mathbf{A}) \leq 1$ par un ensemble de contraintes linéaires. La solution est initialisée avec la solution du problème originel (4.21). Siddiqui et al. ont proposé cette méthode pour l'estimation des paramètres d'un filtre de Kalman appliqué à la prédiction de séries temporelles (ce qui corresponds à $\mathbf{B} = 0$ et $\mathbf{D} = 0$ dans le modèle (4.1)). Dans ce cas, la solution du problème de moindres carrés est mis sous la forme :

$$\mathbf{A}_{CG} = \underset{\mathbf{A}}{\operatorname{argmin}} \left\| \hat{\mathbf{X}}_{t+1} - \mathbf{A}\hat{\mathbf{X}}_t \right\|_F^2$$

Ce problème peut également se mettre sous la forme :

$$\mathbf{a}_{CG} = \underset{\mathbf{a}}{\operatorname{argmin}} (\mathbf{a}^\top \mathbf{H}\mathbf{a} - 2\mathbf{q}^\top \mathbf{a} + r) \tag{4.24}$$

avec ces conventions :

$$\begin{array}{ll} a = vec(\mathbf{A}) & \mathbf{H} = \mathbf{I}_n \otimes (\hat{\mathbf{X}}_t \hat{\mathbf{X}}_t^\top) \\ q = vec(\hat{\mathbf{X}}_t \hat{\mathbf{X}}_{t+1}^\top) & r = \operatorname{trace}(\hat{\mathbf{X}}_{t+1} \hat{\mathbf{X}}_{t+1}^\top) \end{array}$$

où $vec(\mathbf{M})$, $\mathbf{M} \in \mathbb{R}^{n \times n}$ est donné par $vec(\mathbf{M}) = [\mathbf{M}_{11}\mathbf{M}_{21}\mathbf{M}_{31}\ldots\mathbf{M}_{nn}]$ et le symbole $\otimes$ représente le produit de Kronecker.

À chaque étape k, le problème (4.24) est résolu avec une contrainte linéaire sur la valeur singulière maximale de la solution courante $\mathbf{A}_k \equiv \mathbf{a}_k$. Cette contrainte linéaire est donnée par :

$$\sigma_{\max}(\mathbf{A}_k) \leq 1 \Rightarrow \operatorname{trace}(\mathbf{u}^\top \mathbf{A}_k \mathbf{v}) \leq 1 \Rightarrow \operatorname{trace}(\mathbf{v}\,\mathbf{u}^\top \mathbf{A}_k) \leq 1$$

et peut être mis sous la forme :

$$\mathbf{g}^\top \mathbf{a}_k \leq 1 \quad \text{avec} \quad \mathbf{g} = vec(\mathbf{v}\,\mathbf{u}^\top) \tag{4.25}$$

$\mathbf{u}$ et $\mathbf{v}$ sont respectivement les vecteurs singuliers à gauche et à droite de la matrice $\mathbf{A}_k$. L'extension de cette méthode aux systèmes dynamiques avec des entrées est simple. Le problème (4.21) devient :

$$[\mathbf{A}_{CG},\ \mathbf{B}_{CG}] = \underset{\mathbf{A},\mathbf{B}}{\operatorname{argmin}} \left\| [\mathbf{A}\ \mathbf{B}] \begin{bmatrix} \hat{\mathbf{X}}_t \\ \hline \mathbf{U}_{t|t} \end{bmatrix} - \hat{\mathbf{X}}_{t+1} \right\|_F^2$$

sous la contrainte (4.25). Le problème quadratique correspondant s'écrit :

$$\mathbf{a}_{CG} = \underset{\mathbf{a}}{\operatorname{argmin}} (\mathbf{a}^\top \mathbf{H}\mathbf{a} - 2\mathbf{q}^\top \mathbf{a} + r) \tag{4.26}$$

avec les conventions suivantes :

$$\begin{array}{ll} \mathbf{a} = vec([\mathbf{A}\ \mathbf{B}]) & \mathbf{H} = \mathbf{I}_n \otimes \left(\begin{bmatrix} \hat{\mathbf{X}}_t \\ \hline \mathbf{U}_{t|t} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{X}}_t \\ \hline \mathbf{U}_{t|t} \end{bmatrix}^\top \right) \\ \mathbf{q} = vec\left(\begin{bmatrix} \hat{\mathbf{X}}_t \\ \hline \mathbf{U}_{t|t} \end{bmatrix} \hat{\mathbf{X}}_{t+1}^\top \right) & r = \operatorname{trace}\left(\hat{\mathbf{X}}_{t+1} \hat{\mathbf{X}}_{t+1}^\top \right) \end{array}$$

Le vecteur $\mathbf{g}$ utilisé dans (4.25) est donc à chaque étape $\mathbf{g} = vec\left(\left[\mathbf{v}\,\mathbf{u}^\top\ 0_{n \times l} \right] \right)$. La matrice $\mathbf{G}$ est définie par :

$$\mathbf{G} = \begin{bmatrix} \mathbf{g_1}^\top \\ \mathbf{g_2}^\top \\ \mathbf{g_3}^\top \\ \ldots \\ \mathbf{g_k}^\top \end{bmatrix}$$

et l'ensemble des contraintes est défini à chaque étape par :

$$\mathbf{Ga} \leq \mathbf{1}$$

L'ensemble des contraintes linéaires (figure 4.5) converge vers l'ensemble $\left\{\mathbf{A} \in \mathbf{R}^{n\times n}, \sigma_{\max}(\mathbf{A}) \leq 1\right\}$, mais comme les contraintes sont ajoutées itérativement, l'algorithme s'arrête dès qu'une solution stable est trouvée.

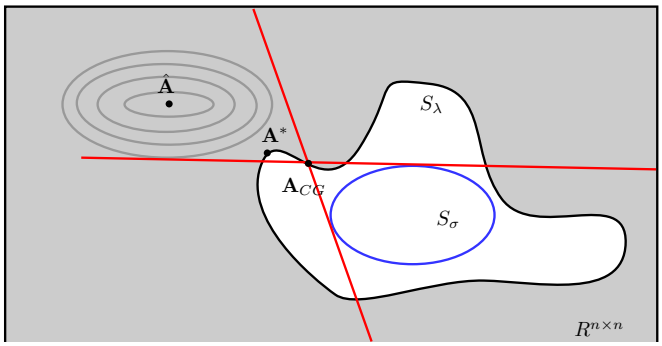


FIGURE 4.5 : Vue imagée des contraintes utilisées pour l'algorithme CG. Les différents ensembles sont définis dans 4.4. En rouge, les contraintes linéaires ajoutées itérativement. $\mathbf{A}_{CG}$ représente la solution trouvée par l'algorithme CG.

4.6 Approches proposées

Nous décrivons ci-après nos approches de solution en tentant d'aller au-delà des contraintes des stabilité convexes employées par Lacy et Bernstein [LB02, LB03] et Siddiqui et al. [SBG08]. La première méthode proposée est une généralisation de l'approche LB1 [LB02] alors que la deuxième méthode optimise directement le problème 4.21 sous la contrainte non-convexe $\rho(\mathbf{A}) \leq 1$.

4.6.1 Contrainte convexe optimale sur la valeur singulière de A (méthodes LBopt)

Bien que l'algorithme de Siddiqi et al. obtienne de bons résultats sur les systèmes avec entrées, quelques questions méritent d'être soulevées. Tout d'abord, il n'y a pas de résultats théoriques sur le nombre d'itérations nécessaires pour la convergence de l'algorithme [SBG08]. Cependant, cette convergence est garantie pour un nombre fini d'itérations. De plus, malgré l'assouplissement des contraintes, la solution obtenue peut être éloignée de la solution optimale $\mathbf{A}^*$ qui minimise J_1 avec $\rho(\mathbf{A}^*) \leq 1$ (voir la figure 4.5). Ceci nous a poussé à chercher un algorithme capable de s'approcher plus finement de la véritable solution sur l'ensemble $\left\{\mathbf{A} \in \mathbf{R}^{n\times n}, \rho(\mathbf{A}) \leq 1\right\}$. Cette analyse est basée sur l'utilisation de l'algorithme LB1, pour lequel les contraintes sont plus facilement interprétables. On sait que $\mathbf{I}_n - \mathbf{A}\mathbf{A}^\top \succeq 0_n$ assure $\sigma_{\max}(\mathbf{A}) \leq 1$ et donc un système stable, mais cette contrainte est trop restrictive. Si l'on considère $\hat{\mathbf{A}}$ la solution du problème non contraint (4.21) et si le système identifié est instable, on a $\rho(\hat{\mathbf{A}}) \geq 1$ et donc $\sigma_{\max}(\hat{\mathbf{A}}) = \delta_{LS}$ avec $\delta_{LS} > 1$ car les valeurs propres de $\hat{\mathbf{A}}$ respectent l'inégalité

$$\sigma_{\min}(\hat{\mathbf{A}}) \leq \rho(\hat{\mathbf{A}}) \leq \sigma_{\max}(\hat{\mathbf{A}})$$

Cependant, entre la solution optimale du problème non contraint $\hat{\mathbf{A}}$ et la solution stable $\mathbf{A}_{LB1}$ telle que $\sigma_{\max}(\mathbf{A}_{LB1}) \leq 1$, il peut exister une meilleure solution. Cette solution doit être stable et respecte $\sigma_{\max}(\mathbf{A}) = \delta$ avec $1 \leq \delta < \delta_{LS}$. Elle correspond à la relaxation de l'inégalité de Lyapunov avec $\mathbf{P} = \mathbf{I}_n$ et $\delta^2\mathbf{I}_n - \mathbf{A}\mathbf{A}^T \succeq 0_n$. La dernière condition implique $\sigma_{\max}(\mathbf{A}) \leq \delta$.

Entrées : $\hat{\mathbf{A}}$, $\hat{\mathbf{B}}$, $\hat{\mathbf{X}}_t$, $\hat{\mathbf{X}}_{t+1}$, $\mathbf{U}_t$, ε
Résultat : $\mathbf{A}_{opt}$, $\mathbf{B}_{opt}$
Initialiser $\delta \leftarrow \sigma_{\max}(\hat{\mathbf{A}})$;
tant que $\rho(\mathbf{A}) > 1$ **faire**
  résoudre $\left\| \hat{X}_{t+1} - [\mathbf{A}\ \mathbf{B}] \begin{bmatrix} \hat{\mathbf{X}}_t \\ \mathbf{U}_{t\|t} \end{bmatrix} \right\|_F^2$;
  sous la contrainte $\delta^2 \mathbf{I}_n - \mathbf{A}\mathbf{A}^\top \succeq 0_n$;
  $\delta \leftarrow \delta - \varepsilon$;
fin
$\mathbf{A}_{LBopt} \leftarrow \mathbf{A}$;
$\mathbf{B}_{LBopt} \leftarrow \mathbf{B}$;

Algorithme 1: Algorithme LBopt, extension de la méthode de Lacy et Bernstein [LB02]

Une manière de trouver la solution optimale stable est donc de trouver la valeur maximale δ telle que le modèle identifié soit stable. On peut remarquer que $\delta_1 < \delta_2$ implique $\{\mathbf{A} \in \mathbf{R}^{n\times n}, \sigma_{max}(\mathbf{A}) < \delta_1\} \subset \{\mathbf{A} \in \mathbf{R}^{n\times n}, \sigma_{max}(\mathbf{A}) < \delta_2\}$. Ainsi en diminuant graduellement δ, on peut espérer trouver la valeur optimale de δ. Cependant, cette solution, nommée LBopt par la suite, nécessite de résoudre plusieurs fois le problème LB1 et est donc prohibitive en terme de calcul. La précision ε est fixée pour décroître δ entre deux itérations. Le nombre d'itérations est donc borné par $N = \left(\frac{\delta_{LS}-1}{\varepsilon}\right)$. Cette extension de la méthode LB1 est présentée dans l'algorithme 1 et la figure 4.6 permet de visualiser le fonctionnement de cette méthode.

L'algorithme 1 est donc capable de trouver une solution en un nombre fini d'itérations (fixé par l'utilisateur) entre la solution des moindres carrés (potentiellement instable) et la solution donnée par LB1. La proximité de la solution avec le minimum local de (4.21) sous la contrainte $\rho(\mathbf{A}) \leq 1$ dépend de la valeur fixée pour ε et de la convexité locale de la fonction $\rho(\mathbf{A}) < 1$. La proximité sera bonne si le pas est faible (et donc le nombre d'itérations important) et si l'ensemble $\{\mathbf{A}, J_1(\mathbf{A},\mathbf{B}) \leq J_1(\mathbf{A}_{LBopt}, \mathbf{B}_{LBopt})\} \cap \{\mathbf{A}, \sigma_{\max}(\mathbf{A}) < \delta_{LBopt}\}$ est convexe. Dans tous les cas, à la fin de l'algorithme, la solution obtenue est au moins meilleure que la solution de LB1 (vis-à-vis du coût de reconstruction).

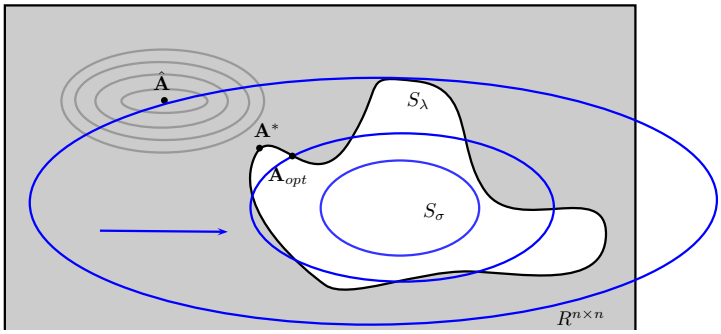


FIGURE 4.6 : Vue imagée des contraintes utilisées pour l'algorithme LBopt. En bleu, plusieurs contraintes convexes du type $\sigma_{\max}(\mathbf{A}) < \delta$ utilisées successivement jusqu'à trouver $\mathbf{A}_{opt}$, solution de l'algorithme LBopt.

4.6.2 Gradient échantillonné (méthode GS)

La solution précédente peut être vue comme une heuristique pour trouver une solution stable qui minimise (4.21). En effet, même si on raffine notre approximation de l'ensemble $\rho(\mathbf{A}) \leq 1$, l'algorithme d'optimisation peut toujours trouver une solution très éloignée de la solution optimale. Ceci est renforcé par le fait que l'ensemble des systèmes asymptotiquement stable est non seulement non-convexe mais également non-lisse. Toutefois, Burke et al. [BLO05] ont proposé un algorithme robuste applicable pour l'optimisation non différentiable et non convexe.

L'algorithme est fondé sur un gradient échantillonné et la seule nécessité est que le gradient de la fonction à optimiser puisse être calculé localement. Les auteurs ont appliqué la méthode pour évaluer la distance de stabilité des matrices ([BLO02]) pour les systèmes linéaires continus. Dans ce cas, la condition de stabilité est uniquement liée à la partie réelle des valeurs propres, ce qui conduit à étudier $\max_i Re(\lambda_i)$.

Le problème d'identification d'une réalisation d'état d'un système discret stable sous contrainte de stabilité s'écrit :

$$\min_{\mathbf{A},\mathbf{B}} J_1 = \left\| \mathbf{X}_{t+1} - [\mathbf{A}\ \mathbf{B}] \begin{bmatrix} \hat{\mathbf{X}}_t \\ \mathbf{U}_{t|t} \end{bmatrix} \right\|_F^2 \text{ s.c. } \quad \rho(\mathbf{A}) \leq 1$$

Pour résoudre ce problème, on utilise la méthode d'Uzawa [AHU58]. On considère le lagrangien $\mathscr{L}$ défini par :

$$\mathscr{L}(\mathbf{A},\mathbf{B},\alpha) = J_1(\mathbf{A},\mathbf{B}) + \alpha(\rho(\mathbf{A}) - 1)$$

avec $\alpha \geq 0$ le paramètre de Lagrange.

Dans notre cas, le problème d'optimisation se résout en itérant les deux étapes :

$$\begin{aligned} \left[\mathbf{A}^{k+1}, \mathbf{B}^{k+1}\right] &= \min_{\mathbf{A},\mathbf{B}} \mathscr{L}(\mathbf{A}^k, \mathbf{B}^k, \alpha^k) \\ \alpha^{k+1} &= \max\left(0, \alpha^k + \beta\left(\rho(\mathbf{A}) - 1\right)\right) \end{aligned}$$

partant d'une solution initiale $\mathbf{A}^0$, $\mathbf{B}^0$ et α^0 jusqu'à convergence. Dans la dernière équation $\beta > 0$ représente le pas du gradient.

On remarque que pour toute matrice $\mathbf{A}$, la matrice $\mathbf{B}$ optimale peut être retrouvée directement par moindres carrés puisque $\mathbf{B}$ n'est pas concerné par la contrainte. Pour utiliser un algorithme de descente de gradient par rapport à $\mathbf{A}$ sur la première équation de ce problème d'optimisation, on doit donc uniquement calculer :

$$\nabla_{\mathbf{A}} \mathscr{L}(\mathbf{A},\mathbf{B},\alpha) = \nabla_{\mathbf{A}} J_1(\mathbf{A},\mathbf{B}) + \alpha \nabla_{\mathbf{A}} \rho(\mathbf{A})$$

Les différents paramètres sont ensuite adaptés itérativement selon la règle :

$$\begin{aligned} \mathbf{A}^{t+1} &= \mathbf{A}^t - \eta \nabla_{\mathbf{A}} \ell(\mathbf{A}^t, \mathbf{B}^t) \\ \mathbf{B}^{t+1} &= \min_{\mathbf{B}} J_1(\mathbf{A}^{t+1}, \mathbf{B}^t) \end{aligned}$$

où $\eta > 0$ est le pas du gradient et l'estimation $\mathbf{B}^{t+1}$ est simplement obtenue par moindres carrés.

Le calcul de $\nabla_{\mathbf{A}} J_1(\mathbf{A},\mathbf{B})$ est assez simple. En effet, on a à partir des formules 121 et 108 de [PP08] :

$$\frac{\partial}{\partial \mathbf{X}} \|\mathbf{X}\|_F^2 = \frac{\partial}{\partial \mathbf{X}} \text{trace}\left[\mathbf{X}\mathbf{X}^\top\right]$$

$$\frac{\partial}{\partial \mathbf{X}} \text{trace}\left[(\mathbf{AXB}+\mathbf{C})(\mathbf{AXB}+\mathbf{C})^\top\right] = 2\mathbf{A}^\top(\mathbf{AXB}+\mathbf{C})\mathbf{B}^\top$$

et l'on obtient :

$$\begin{aligned} \nabla_{\mathbf{A}} J_1(\mathbf{A},\mathbf{B}) &= \nabla_{\mathbf{A}} \left\|\hat{\mathbf{X}}_{t+1} - \mathbf{A}\hat{\mathbf{X}}_t - \mathbf{B}\mathbf{U}_{t|t}\right\|_F^2 \\ &= \nabla_{\mathbf{A}} \text{trace}\left[\left(\hat{\mathbf{X}}_{t+1} - \mathbf{A}\hat{\mathbf{X}}_t - \mathbf{B}\mathbf{U}_{t|t}\right)^\top \left(\hat{\mathbf{X}}_{t+1} - \mathbf{A}\hat{\mathbf{X}}_t - \mathbf{B}U_{t|t}\right)\right] \\ &= -2\hat{\mathbf{X}}_t^\top \left(\hat{\mathbf{X}}_{t+1} - \mathbf{A}\hat{\mathbf{X}}_t - \mathbf{B}\mathbf{U}_{t|t}\right) \end{aligned}$$

Pour calculer $\nabla_{\mathbf{A}}\mathscr{L}(\mathbf{A},\mathbf{B})$, on a besoin aussi de calculer le gradient de $\rho(\mathbf{A})$. Soit $\mathbf{u}_1$ et $\mathbf{v}_1$ les vecteurs propres à droite et à gauche de $\mathbf{A}$(sous la contrainte $\mathbf{u}_1^\top \mathbf{v}_1 = 1$). On peut montrer que ([RH85], Théorème 6.3.12]) si λ_1 est une valeur propre simple réelle, ce gradient est défini par :

$$\nabla_{\mathbf{A}}\rho(\mathbf{A}) = \text{sign}(\lambda_1)\mathbf{u}_1\mathbf{v}_1^\top$$

Quand le rayon spectral est associé à une paire conjuguée de valeurs propres $(\lambda_1 \pm i\tilde{\lambda}_1)$, on doit considérer séparément la partie réelle et la partie imaginaire des vecteurs propres ($\mathbf{u}_1 \pm i\tilde{\mathbf{u}}_1$ et $\mathbf{v}_1 \pm i\tilde{\mathbf{v}}_1$). Comme les gradients des valeurs propres sont conjugués, on peut écrire :

$$\nabla_{\mathbf{A}}\rho(\mathbf{A}) = \frac{\lambda_1(\mathbf{u}_1\mathbf{v}_1^\top + \tilde{\mathbf{u}}_1\tilde{\mathbf{v}}_1^\top) - \tilde{\lambda}_1(\tilde{\mathbf{u}}_1\mathbf{v}_1^\top - \mathbf{u}_1\tilde{\mathbf{v}}_1^\top)}{|\lambda_1 \pm i\tilde{\lambda}_1|}$$

Toutefois, comme la fonction $\rho(\mathbf{A})$ est non convexe et non différentiable, il convient d'utiliser l'algorithme du gradient échantillonné(algorithme 2) pour le calcul du gradient du rayon spectral de $\mathbf{A}$. Les détails sur la sous procédure de recherche linéaire et la convergence de l'algorithme peuvent être trouvés dans [BLO05].

Entrées : $\mathbf{A}$, ε, N, $\theta < 1$, t_{max}
Résultat : $\nabla_{\mathbf{A}}\rho(\mathbf{A})$
Tirer $N-1$ matrices $\mathbf{A}_j$ au voisinage de $\mathbf{A}$ contrôlé par le rayon d'échantillonnage ε
Initialiser $G = \{\nabla_{\mathbf{A}}\rho(\mathbf{A}_j)\}$
Calculer la direction de recherche d par la minimisation quadratique convexe $d = \text{argmin}_{g\in\text{convex}(G)}(\|g\|^2)$
si $d = 0$ **alors**
  Décroitre ε : $\varepsilon \leftarrow \theta\varepsilon$
  Recommencer
sinon
  Utiliser la recherche linéaire monodimensionnelle pour trouver $t > 0$ tel que $\rho(\mathbf{A}_k + td) < \rho(\mathbf{A}_k)$ avec $0 < t \leq t_{max}$
  $\nabla_{\mathbf{A}}\rho(\mathbf{A}) \leftarrow td$
fin

Algorithme 2: Algorithme de gradient échantillonné

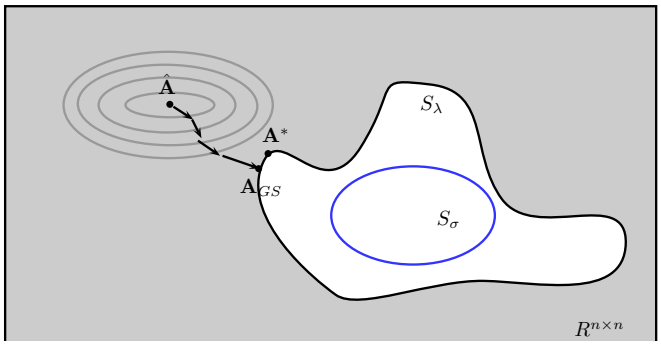


FIGURE 4.7 : Vue imagée des contraintes utilisées pour l'algorithme GS. Chaque flèche représente une itération de descente de gradient vers la solution finale $\mathbf{A}_{GS}$

4.7 Expériences

Nous avons ensuite appliqué ces différentes méthodes LB1 (4.5.1), CG (4.5.2), LBopt (4.6.1) et GS (4.6.2) sur un exemple numérique, LB2 4.5.1 n'ayant pas été retenu à cause d'une trop grande erreur de reconstruction. On

$$\left[\begin{array}{c|c} \mathbf{A} & \mathbf{B} \\ \hline \mathbf{C} & \mathbf{D} \end{array}\right] = \left[\begin{array}{cccccccc|cc} 0 & 0 & 0 & 0 & 0 & 0 & 0.5 & 0.5 & -1.1281 & 1.8120 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1.0713 & 0.6069 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0.0237 & -0.9395 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0.4678 & -0.1840 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0.7559 & 0.4297 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0.0845 & 0.5869 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1.3911 & -1.1763 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0.3262 & 0.4888 \\ \hline 0.5312 & -1.3323 & -0.1252 & 0.7311 & 1.5951 & 0.9462 & -0.2494 & 0.6099 & -1.1218 & 0.7253 \\ -0.4550 & -1.7670 & 0.0868 & -0.5294 & 1.5878 & -1.8315 & 1.0805 & 2.4086 & -1.8313 & -0.3749 \end{array}\right]$$

TABLE 4.2 : Matrices du système simulé

	LS	LB1	CG	LBopt	GS1	GS2
$\rho(\mathbf{A})$	1.035	0.9835	0.9999	0.9999	1	1
$\sigma_{max}(\mathbf{A})$	1.0610	1	1.0527	1.0482	1.0602	1.0603
ex. time	0.0410	12.9888	0.0825	35.55387	0.8013	6.7002
$J_1(\mathbf{A},\mathbf{B})$	0.2945	0.4865	0.3259	0.3116	0.2948	0.2947
$\left\|\hat{\mathbf{Y}}_{1\|t} - \mathbf{Y}_{1\|t}\right\|_F$	Inf	540.95	93.35	140.66	41.93	41.30

TABLE 4.3 : Comparaison des performances moyennes sur les 50 jeux de données des différents algorithmes.

ajoute un bruit blanc **w** de moyenne nulle à la sortie du système simulé, tel que le rapport signal/bruit soit égal à :

$$S/N = \left\|\mathbf{Y}_{1|t} - \mathbf{W}_{1|t}\right\|_F / \left\|\mathbf{W}_{1|t}\right\|_F = 10$$

Les données d'entrée sont générées comme un signal aléatoire gaussien. L'algorithme N4SID [OM94] est utilisé pour identifier la séquence d'états. Chaque algorithme est testé 50 fois avec différents jeux d'entrées/sortie, en éliminant les cas où le système identifié est directement stable. L'ordre réel du système ($n = 8$) est donné en entrée des différents algorithmes. ε est fixé à 0.01 pour LBopt. Pour GS, ε est fixé 0.001, θ à 0.8 et t_{max} à 0.1. De plus, le cas présenté dans [MGC08] correspond à un α^0 infini. Ce cas sera nommé GS1. Une autre initialisation de l'algorithme a été faite avec $\alpha^0 = 0$ et $\beta = 100$. Ce cas sera nommé GS2.

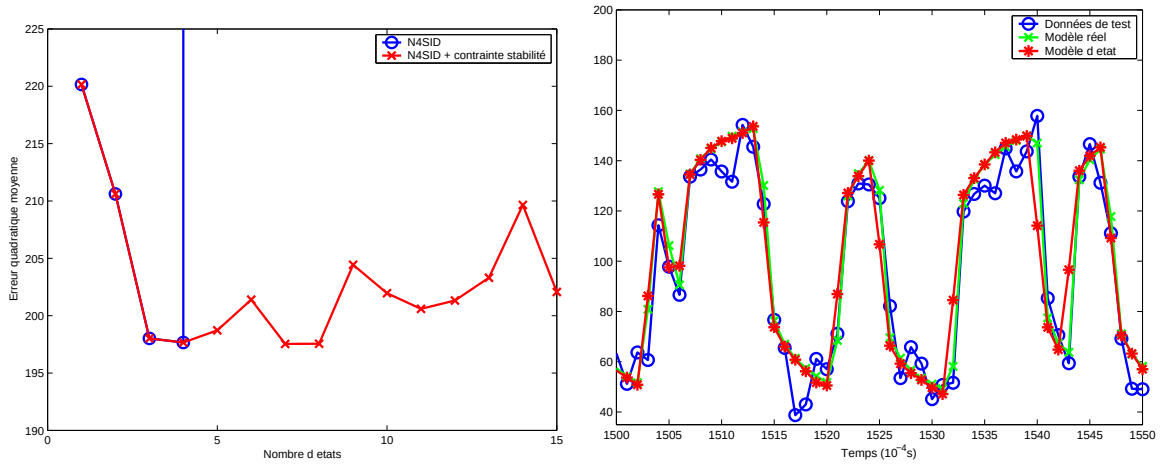
L'exemple utilisé est inspiré de [LB03]. Les paramètres du système sont donnés par la table 4.2. Nous avons choisi $\rho(\mathbf{A}) = 1$ pour avoir un système réel stable mais aussi pour qu'avec du bruit ou un nombre de données insuffisant, la solution des moindres carrés soit instable. Les performances des différents algorithmes sont comparées à partir de l'erreur de reconstruction ($J_1(\mathbf{A},\mathbf{B})$ 4.20) sur l'ensemble d'apprentissage et l'erreur de simulation ($\left\|\hat{\mathbf{Y}}_{1|t} - \mathbf{Y}_{1|t}\right\|_F$) sur l'ensemble de test. Le rayon spectral $\rho(\mathbf{A})$, la plus grande valeur singulière, notée $\sigma_{\max}(A)$, ainsi que le temps d'exécution sont également indiqués, comme le montre le tableau 4.3.

Les résultats sur le rayon spectral et la valeur singulière maximale sont conformes aux attentes. Pour l'erreur de reconstruction, il est notable que les performances des 2 algorithmes de gradient échantillonné sont très proches du minimum atteignable. Comme prévu, LB1 obtient les moins bons résultats, du fait d'un contrainte trop restrictive. Les résultats des algorithmes CG et LBopt sont similaires sur cet indicateur. Concernant l'erreur sur l'ensemble de test, comme nous avons volontairement sélectionné des modèles instables, ce critère n'est pas pertinent pour les modèles identifiés uniquement en minimisant $J_1(\mathbf{A},\mathbf{B})$. Les rangs de CG et LBopt sont inversés sur ce critère et GS (GS1 et GS2) obtient les meilleures performances. Dans ce cas, les contraintes linéaires semblent agir comme un paramètre régularisateur, ce qui permet à l'algorithme CG d'obtenir de meilleures performances que LBopt qui lui reste sur une solution trop proche de celle des moindres carrés. Le temps d'exécution de CG est clairement le plus court parmi les méthodes proposées. Pour l'algorithme GS, on constate que le cas GS1 est beaucoup plus rapide que le cas GS2, même si cela se paie par des performances légèrement inférieures.

4.8 Applications aux données thermiques

L'algorithme N4SID [OM94] est utilisé à nouveau pour identifier les modèles d'états. L'apprentissage est fait avec et sans la contrainte de stabilité sur la matrice **A**. Le seul paramètre restant à choisir est le nombre d'états du système. On compare donc l'erreur quadratique moyenne obtenue pour les différentes architectures en utilisant la procédure de validation croisée décrite pour les algorithmes du chapitre précédent.

Pour les données « Transistor », on constate que le meilleur résultat est réalisé pour un modèle d'ordre 7 (figure 4.8(a)). Ce résultat a pû être obtenu grâce à l'utilisation de la contrainte de stabilité puisque que l'on constate que dans le cas non-contraint, des modèles instables ont été identifié à partir de l'ordre 4.La sortie obtenue est fortement similaire à celle obtenue pour le modèle linéaire (figure 4.8(b)) même si l'on constate une légère différence de l'erreur quadratique moyenne en apprentissage en faveur du modèle linéaire OE($J_{val} = 197.539$) et sur la base de test en faveur du modèle d'état $J_{test} = 185.683$ pour les données avec bruit mais pas pour les données sans bruit $J_{test} = 80.327$ (voir tableau 3.4 pour les performances des modèles linéaires).



(a) Variation de l'erreur de validation en fonction du nombre d'états du système.

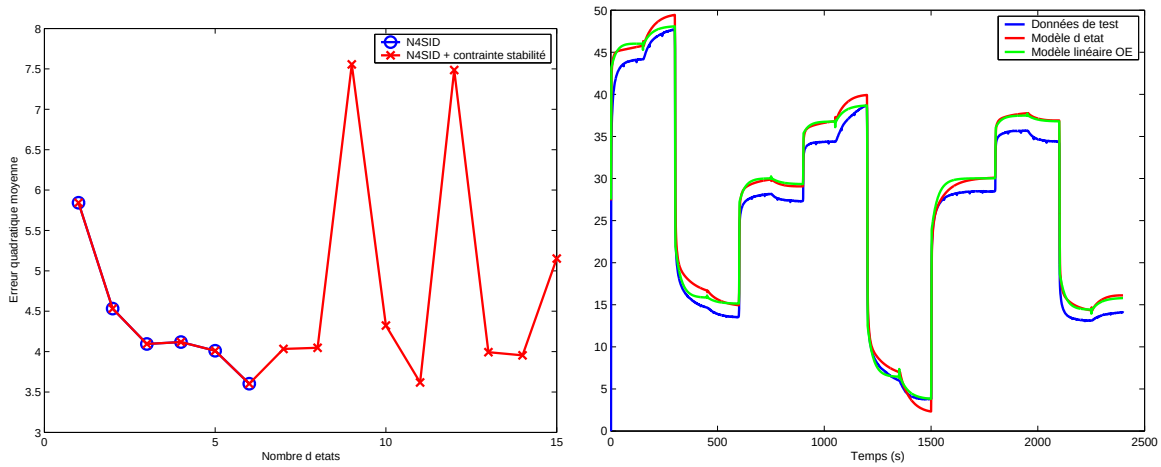
(b) Comparaison de la sortie du modèle d'état avec les données « Transistor » avec et sans bruit.

FIGURE 4.8 : Résultats obtenus pour un modèle d'états sur la base de données « Transistor »

Pour les données « Maquette », on constate que le meilleur résultat est réalisé pour un modèle d'ordre 6 (figure 4.9(a)). Cette fois-ci, le résultat optimal est le même avec ou sans l'utilisation de la contrainte de stabilité. Toutefois, la contrainte de stabilité permet de confirmer que le modèle d'ordre 6 est le plus performant, puisque comme dans le cas « Transistor », au-delà de cette limite, les modèles obtenus par la méthode n4sid sans contrainte sont instables. La sortie obtenue est comparée à celle obtenue pour le modèle linéaire (figure 4.9(b)). On constate une différence de l'erreur quadratique moyenne en apprentissage ($J_{val} = 3.603$) et sur la base de test $J_{test} = 3.5579$ en faveur du modèle linéaire OE (voir tableau 3.5 pour les performances des modèles linéaires).

4.9 Conclusion et perspectives

Après une analyse des différentes méthodes existantes pour l'identification d'un modèle linéaire stable pour les systèmes dynamiques, nous avons constaté que toutes ces méthodes se fondaient sur une approximation de la contrainte de stabilité par une contrainte convexe. De plus, moins la contrainte utilisée est restrictive, meilleures sont les performances de la méthode. Ce fait est la motivation principale pour proposer une méthode qui optimise directement le critère des moindres carrés sur l'erreur de reconstruction avec une contrainte sur le rayon spectral de la matrice A. Le problème est non-convexe et non-différentiable mais peut être résolu en utilisant un algorithme de gradient échantillonné. Les résultats obtenus sur des données simulées et sur les données thermiques montrent



(a) Variation de l'erreur de validation en fonction du nombre de d'états du système.

(b) Comparaison de la sortie du modèle d'état avec les données « Maquette ».

FIGURE 4.9 : Résultats obtenus pour un modèle d'états sur la base de données « Maquette »

la pertinence de cette démarche. Il serait alors pertinent d'étendre nos approches à des cas non-linéaires.

Une extension non-linéaire simple du modèle (4.1) consiste à adopter une approche à base de noyaux [SS01]. L'idée est de projeter de façon non-linéaire les entrées $u(t)$ et les sorties $y(t)$ dans un espace $\mathscr{H}$. Dans cet espace, on recherche un modéle d'état de la forme [RdB04, KYM07] :

$$\left\{ \begin{array}{lcl} \mathbf{X}_\phi(t+1) & = & \mathbf{A}_\phi \mathbf{X}_\phi(t) + \mathbf{B}_\phi \phi_\mathbf{u}(\mathbf{u}(t)) \\ \phi_\mathbf{y}(\mathbf{y}(t)) & = & \mathbf{C}_\phi \mathbf{X}_\phi(t) + \mathbf{D}_\phi \phi_\mathbf{u}(\mathbf{u}(t)) \end{array} \right.$$

avec $\phi_\mathbf{u}(.)$ et $\phi_\mathbf{y}(.)$ les fonctions de projection non-linéaires. Connaissant la sortie $\phi_\mathbf{y}(\mathbf{y}(t))$ dans $\mathscr{H}$, on détermine l'estimation $\hat{y}$ en faisant une projection inverse (ou du moins, trouver une approximation si la projection n'est pas bijective). Pour réaliser cette transformation inverse, on peut par exemple utiliser la formulation dans [KYM07]. L'avantage de cette approche est que l'adaptation des contraintes de stabilité est immédiate. En effet, une fois la séquence d'états identifiée, les cas linéaire et non-linéaire peuvent être traités de façons identiques. On pourrait alors déterminer des modèles d'état non-linéaires avec des contraintes de stabilité.

5

Conclusion

Dans ces travaux, nous nous sommes intéressés à la modélisation thermique appliquée aux systèmes RADAR. En effet, le développement de modèles thermiques plus performants est une nécessité pour permettre de relever des défis toujours plus importants en termes de puissance d'émission, de conditions extérieures et de système de refroidissement utilisé. Dans le développement de modèles thermiques, nous nous sommes principalement intéressés aux méthodes statistiques capables de produire des modèles compacts à partir de données. Le but à terme est de pouvoir embarquer ces modèles sur des composants présents à l'intérieur du RADAR.

La génération des données nécessaires pour permettre l'apprentissage statistique nous a amené à utiliser les modèles classiquement étudiés en modélisation thermique, tels que les modèles à éléments finis. Cette étape nous a permis de constater la difficulté de réaliser un modèle complet de l'architecture du système. En effet, en partant des doigts des transistors de puissance (de quelques centaines de micromètres) jusqu'au système de refroidissement (qui atteignent des dizaines de centimètres), la plage des temps de réponse thermique est bien trop importante pour la réalisation de modèles et même de bases de données exploitables. Cette étape nous a donc conduit à séparer la modélisation thermique en 2 problèmes distincts, selon que l'on s'intéresse à la température du composant ou à celle du système de refroidissement. De plus, elle nous a permis de mieux comprendre les non-linéarités qui pourraient être présentes dans les données thermiques.

Pour générer d'autres données, les méthodes de mesures thermiques ont également été utilisées. Pour la température du composant, les fortes exigences en termes de temps de réponse et de précision nous ont conduit à retenir les mesures au microscope infrarouge. Pour la température du système de refroidissement, les thermocouples forment le meilleur compromis en terme de performances et de praticité. En combinant ces deux étapes de modélisation et de mesure thermiques, deux séries de données ont été créées afin d'évaluer les performances des méthodes d'apprentissage statistiques pour la prédiction de température dans les systèmes RADAR.

Plusieurs méthodes d'apprentissage statistique, linéaires et non-linéaires, de modèles dynamiques ont été étudiées. Parmi celles existantes, nous nous sommes plus particulièrement intéressés aux modèles récurrents, compte tenu de la forte inertie présente dans les données thermiques et du fait qu'aucun système de mesure ne pourra être embarqué dans un RADAR. Toutes nos expériences ont d'ailleurs montré que, pour une classe de modèles donnée, les performances étaient améliorées par la prise en compte de cette récurrence (sauf pour les Least Square SVM récurrents qui constituent ainsi un cas particulier). Les méthodes récurrentes doivent résoudre deux problèmes simultanément. Tout d'abord, la minimisation de l'erreur vis-à-vis de la base d'apprentissage (ce problème est commun à toutes les méthodes d'apprentissage). Mais, pour un modèle récurrent, il faut également vérifier sa capacité à maintenir ces performances en utilisant des données de sortie qu'il a lui même générées. Ces deux étapes sont présentes dans tous les algorithmes récurrents que nous avons étudiés, de manières implicites pour les modèles linéaires, les réseaux de neurones (lors de la mise à jour des paramètres du modèle au cours de l'optimisation) et RLSSVM (où cette étape est intégrée dans les contraintes du problème d'optimisation) ou plus explicites pour les

réseaux bayésiens (les deux étapes de l'algorithme EM) ou les modèles d'états (projection, puis minimisation de l'erreur de reconstruction sur la séquence d'états), de manière itérative (les modèles linéaires, les réseaux de neurones, RLSSVM et réseaux bayésiens) ou non (modèle d'états). Pour les modèles d'états, il est donc important que les hypothèses utilisées lors de la projection vers l'espace d'état soient vérifiées, sous peine d'obtenir une séquence d'états impossible à reconstruire.

Au niveau des performances, les modèles linéaires présentent déjà des résultats satisfaisants pour les 2 séries de mesures. Les non-linéarités théoriques des modèles thermiques n'ont donc qu'une faible influence. Pour la modélisation du système de refroidissement, l'utilisation d'un réseau de neurones peut être envisagée si les performances attendues justifient une complexité de mise en oeuvre plus importante. Les autres méthodes non linéaires ne semblent pas adaptées à ce problème, et les réseaux bayésiens n'apportent pas de différences significatives vis-à-vis des modèles linéaires. Pour les méthodes à noyau, les méthodes de sous-espaces semblent être une alternative intéressante aux machines à vecteurs supports dans le cadre d'un modèle dynamique.

Pour les modèles récurrents, lors de l'étape consistant à vérifier sa capacité à maintenir les performances pour des données de sortie qu'il a lui même générées, l'utilisation de contraintes, notamment de contraintes de stabilité, peut être un atout utile dans la recherche d'un modèle. Nous avons démontré que ce type de contraintes peut être intégré dans un algorithme de sous-espace pour la construction d'un modèle d'états linéaire, améliorant ainsi les performances de la méthode. L'intégration de ces contraintes dans un algorithme de sous-espace non-linéaire utilisant des noyaux peut être une piste intéressante pour l'apprentissage de modèles dynamiques. Leur application dans ce cadre non-linéaire aux données thermiques reste à mener.

Pour continuer le développement de modèles pour prédire la température dans des systèmes RADAR, l'élément le plus important est la création de bases de données plus adaptées à l'apprentissage statistique. La réalisation d'un banc de mesure thermique pour les composants électroniques qui soit capable de générer des données d'entrée variées est donc nécessaire. De même, la commande de la vitesse de l'air dans le système de refroidissement de la maquette CORIA doit être automatisée pour permettre de créer des données plus intéressantes. Au niveau des modèles statistiques, sur les données disponibles, les performances obtenues sont satisfaisantes pour l'application envisagée. De nouvelles données permettraient cependant de confirmer ces résultats. Les modèles d'états sont une approche intéressante pour l'apprentissage de nouveaux modèles thermiques, notamment si plusieurs points de mesure sont présents sur le même système.

A

Repositionnement et recalage d'images infrarouges

(19) RÉPUBLIQUE FRANÇAISE

INSTITUT NATIONAL
DE LA PROPRIÉTÉ INDUSTRIELLE

PARIS

(11) **N° de publication :** **2 938 100**
(à n'utiliser que pour les commandes de reproduction)

(21) **N° d'enregistrement national :** **08 06074**

(51) Int Cl[8] : ***G 06 T 3/00*** *(2006.01), H 04 N 1/387, 5/33, G 01 K 17/00, G 01 R 31/308*

(12) DEMANDE DE BREVET D'INVENTION A1

(22) **Date de dépôt :** 31.10.08.

(30) **Priorité :**

(43) **Date de mise à la disposition du public de la demande :** 07.05.10 Bulletin 10/18.

(56) **Liste des documents cités dans le rapport de recherche préliminaire :** *Se reporter à la fin du présent fascicule*

(60) **Références à d'autres documents nationaux apparentés :**

(71) **Demandeur(s) :** *THALES Société anonyme* — FR.

(72) **Inventeur(s) :** POLAERT HUBERT, EUDELINE PHILIPPE et MALLET GREGORY.

(73) **Titulaire(s) :** THALES Société anonyme.

(74) **Mandataire(s) :** MARKS & CLERK FRANCE.

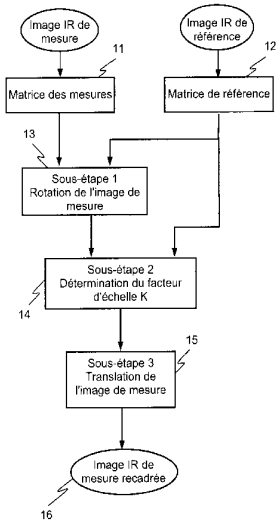
(54) PROCEDE DE RECADRAGE AUTOMATIQUE D'UNE IMAGE INFRAROUGE.

(57) L'invention se rapporte au domaine général du traitement d'images. Elle concerne plus particulièrement le recadrage d'images, d'images infrarouges notamment, en vue de l'exploitation des informations contenues dans ces images par comparaison avec des images de référence.

L'invention consiste en un procédé permettant de réaliser automatiquement ce recadrage, procédé qui comporte trois étapes:

- une étape de rotation qui permet d'aligner les axes de référence d'une image à recadre sur les axes correspondant de l'image de référence associée;
- une étape de mise à l'échelle de l'image à recadrer par rapport à l'image de référence la mise à l'échelle étant réalisée sur l'image après rotation;
- une étape de translation de façon à réaliser la superposition de l'image remise à l'échelle et de l'image de référence.

Le procédé peut être appliqué à la détermination de la température réelle en tout point d'un circuit électronique en fonctionnement, à partir de l'image infrarouge de ce circuit.



FR 2 938 100 - A1

PROCEDE DE RECADRAGE AUTOMATIQUE D'UNE IMAGE INFRAROUGE

L'invention se rapporte au domaine général du traitement d'images. Elle concerne plus particulièrement le recadrage d'images, d'images infrarouges notamment, en vue de l'exploitation des informations contenues dans ces images par comparaison avec des images de référence.

Dans le cadre de l'analyse du bon fonctionnement de circuits électroniques, les mesures de température effectuées sur ces circuits pendant différentes phases de leur fonctionnement permettent de diagnostiquer des défauts de fonctionnement consécutifs par exemple à des erreurs de conception ou de réalisation. La détection d'un "point chaud" sur un circuit peut ainsi indiquer qu'un des composants du circuit se trouve dans une zone de fonctionnement potentiellement destructrice. Elle peut par exemple révéler un problème systématique de réalisation de série, non décelé lors de la conception, ni même lors de la réalisation d'un prototype.

Parmi les méthodes d'analyse utilisées une méthode connue consiste à utiliser l'imagerie infrarouge (IR) pour détecter d'éventuels points chauds au travers de leur signature infrarouge. L'image thermographique d'un circuit apparaît alors comme un ensemble de zones de différentes couleurs, la couleur d'une zone étant fonction de la température du circuit, ou plus exactement de l'énergie rayonnée par cette zone.

L'analyse infrarouge du bon fonctionnement d'un circuit après fabrication, est généralement réalisée alors que celui-ci est mis en test (test in situ ou test fonctionnel). Une image IR du circuit est capturée et utilisée pour déterminer la température en chaque point du circuit.

Pour réaliser une mesure précise et significative à partir de l'image infrarouge, il est nécessaire de prendre en compte l'émissivité de chaque élément constituant le circuit, substrat, composants ou pistes conductrices. En effet pour une température donnée chaque composant émet une onde infrarouge dont l'intensité est principalement fonction de la nature de sa surface, de sorte que deux composants portés à des températures différentes, l'une étant par exemple excessive, peuvent émettre des ondes

infrarouges de même intensité. Par suite sans connaître l'émissivité correspondant à chaque composant il est impossible de déterminer si l'un de ces composants présente une température excessive. Autrement dit, il n'est pas possible de donner une mesure précise de la température d'un composant donné ou encore d'une zone donnée du circuit sans déterminer l'émissivité correspondante. En outre la détermination d'une émissivité moyenne ne permet pas une mesure précise de la température locale dans une zone donnée du circuit à partir d'une image infrarouge.

Déterminer cette émissivité a priori, par le calcul par exemple, s'avère malheureusement impossible. On a donc généralement recours à des opérations de calibration afin d'en effectuer l'estimation. Dans ce contexte, l'approche classique consiste à calculer, à partir d'un circuit pris comme référence, l'émissivité moyenne sur des zones homogènes et à associer les résultats obtenus aux zones correspondantes de l'image infrarouge du circuit analysé. A cet effet, le circuit de référence est placé, hors fonctionnement, dans un environnement porté à une température stabilisée donnée (enceinte climatique). Une image infrarouge est alors réalisée puis les données extraites de l'image sont analysées.

La caméra IR est étalonnée sur un corps noir et n'est sensible qu'aux luminances, de sorte que bien que le circuit soit porté de manière uniforme à une température donnée, la caméra infrarouge délivre une image présentant des zones ayant apparemment des températures différentes. Par suite, connaissant la température à laquelle est porté le circuit, on détermine l'émissivité de chaque zone homogène.

Dans un processus de test automatique, les données de température fournies par la caméra infrarouge sont généralement mémorisées sous forme de pixels, la température associée à chaque pixel étant mémorisée dans une table et référencée par la position du pixel sur l'image. Par suite, l'étalonnage consiste à déterminer la valeur d'émissivité à associer à chaque zone du circuit présentant une température apparente (mesurée par la caméra) homogène, connaissant la position du circuit de référence par rapport à la caméra lors de la réalisation de l'image infrarouge de référence ainsi que la température d'étalonnage à laquelle le circuit est porté. Les valeurs d'émissivité ainsi calculées sont alors utilisées pour déterminer sur chaque

circuit mis en test, les températures respectives des zones correspondant aux zones déterminées à partir de l'image du circuit de référence. Pour ce faire une image infrarouge du circuit sous test est réalisée et les pixels formant cette image, ou plus exactement les zones de l'image homogènes en température, sont associés aux valeurs d'émissivité déterminées lors de l'étalonnage. Cette association nécessite une identification de chaque zone de l'image de mesure à la zone correspondante de l'image de référence.

Ce type d'automatisation se heurte à un problème technique de taille qui consiste à associer de manière automatique la bonne valeur d'émissivité estimée sur le circuit de référence à chacune des zones du circuit analysé. Or dans l'environnement de test, le circuit et l'appareil de prise de vue sont agencés l'un par rapport à l'autre de manière semblable mais non identique à l'agencement d'étalonnage. Il en résulte que le positionnement du circuit par rapport au repère de l'appareil de prise de vue peut être légèrement différent de ce qu'il était lors de l'étalonnage. Par suite, les pixels représentant une même zone sur l'image de référence et sur l'image de test, n'occupent pas toujours des positions identiques sur chaque image de sorte que sans opération de remise en coïncidence, l'association automatique, telle quelle, d'une valeur d'émissivité à une zone de l'image infrarouge du circuit sous test (image de mesure) conduit à une estimation erronée de la valeur de la température de la zone considérée. La remise en coïncidence, doit prendre principalement en compte l'écart, entre la phase d'étalonnage et les phases de test successives, de positionnement du circuit par rapport à la caméra infrarouge. Elle doit également prendre en compte la variation de grandissement pouvant avoir affecté la prise de vue entre ces deux phases.

En l'état actuel de l'art il n'existe pas de moyen permettant d'effectuer de manière entièrement automatique l'ensemble des opérations d'étalonnage et de mesure. En particulier, aucune méthode ne permet de réaliser de manière satisfaisante un recadrage automatique d'une image infrarouge par rapport à une autre On est donc contraint de procéder en plusieurs étapes. La première étape consiste à réaliser l'étalonnage en déterminant sur une image infrarouge de référence les zones homogènes et les valeurs d'émissivité associées. La deuxième étape consiste ensuite à réaliser une image infrarouge du circuit sous test. La troisième étape consiste enfin à

mettre en coïncidence l'image obtenue avec l'image obtenue lors de la phase d'étalonnage. Par mise en coïncidence on entend ici, l'action consistant à déterminer à partir de l'image de référence, la valeur d'émissivité à associer à chacune des zones de l'image de mesure. Les deux premières étapes peuvent être réalisées de manière automatique, tandis que la troisième nécessite généralement l'intervention d'un opérateur. C'est pourquoi dans la pratique, en l'état actuel de l'art, on se contente généralement d'estimer une valeur d'émissivité moyenne à partir de l'image de référence que l'on applique à l'ensemble des zones du circuit. Ceci permet de s'affranchir de la troisième étape et de mettre en œuvre un processus entièrement automatique. En revanche cela rend la mesure de la température locale moins précise et diminue l'efficacité de l'utilisation d'images infrarouges pour déterminer les points chauds.

Un but de l'invention est de rendre possible une automatisation complète d'une mesure précise de la température d'un circuit. Un autre but de l'invention est de proposer un moyen d'assurer le recadrage automatique de deux images d'un même objet prises dans des conditions d'agencement différentes de l'objet et des moyens de prise de vue.

A cet effet l'invention a pour objet un procédé pour réaliser en automatique le recadrage d'une image de mesure infrarouge de la température d'un circuit en fonctionnement sur une image infrarouge de référence réalisée sur un circuit semblable hors fonctionnement et porté à une température uniforme, les deux images infrarouges étant exploitées sous la forme de matrices de pixels. Le procédé selon l'invention comporte les étapes suivantes:

- une première étape durant laquelle on effectue une rotation de l'image de mesure, la valeur de l'angle de rotation étant déterminée par détection des contours des zones de l'image homogènes en température, par détermination de directions prépondérantes dans l'image et par détermination de l'écart angulaire séparant ces directions prépondérantes des directions prépondérantes correspondantes de l'image de référence;

- une deuxième étape durant laquelle on modifie la taille de l'image de mesure réorientée obtenue à l'issue de la sous-étape précédente, la taille de l'image étant modifiée d'un facteur d'échelle K, le facteur K étant déterminé

par appariement des points caractéristiques des deux images, et analyse de l'écart entre la distance des points caractéristiques de l'image de mesure corrigée à une origine donnée et la distance à cette même origine des points caractéristiques de l'image de référence auxquels ils sont appariés;

- une troisième étape durant laquelle l'image obtenue à l'issue de la deuxième sous-étape est translatée de façon à superposer le repère de position de cette image au repère de position de l'image de référence.

Selon un mode de mise en œuvre préféré on effectue, durant la première étape, la détection des contours de l'image de mesure et de l'image de référence par la méthode de Canny.

Selon un mode de mise en œuvre préféré pouvant être combiné avec le précédent, on détermine, durant la première étape, le décalage angulaire des droites prépondérantes en appliquant une transformation de Hough dans un plan (distance, angle de rotation) aux pixels constituant les contours des images et en effectuant une corrélation circulaire entre les vecteurs des maxima de la transformée de hough de chacune des images.

Selon un mode de mise en œuvre préféré pouvant être combiné avec les précédents, les points d'intérêt recherchés lors de la deuxième étape sont des coins déterminés au moyen du détecteur de Harris.

Selon un mode de mise en œuvre préféré, pouvant être combiné avec le précédent, durant la deuxième étape la détermination du facteur d'échelle K est réalisée en découpant l'image de mesure corrigée en rotation et l'image de référence en sous-images, chaque sous-image étant centrée sur un coin et étant caractérisée par ses descripteurs de Hu, puis en effectuant l'appariement des sous images de l'image de mesure avec les sous-images de l'image de référence, l'appariement étant réalisé en calculant les éléments $D_{i,j}$ de la matrice de distance D définie par:

$$D_{i,j} = \sqrt{\sum_{m=1}^{7} \left(\phi_m(i) - \phi_m(j)\right)^2}$$

où i représente la i-ème sous-image de l'image de mesure et j la j-ème sous-image de l'image de référence et $\phi_m(i)$ le moment de Hu d'indice m de la sous-image i.

Selon un mode de mise en œuvre préféré, pouvant être combiné avec les précédents, on effectue durant la troisième étape une corrélation croisée, suivant les deux dimensions, de l'image de mesure corrigée obtenue à l'issue de la deuxième étape et de l'image de référence, de façon à déterminer par rapport à deux directions perpendiculaires l'écart de positionnement existant entre les deux images.

L'invention a également pour objet une application du procédé selon l'invention pour réaliser la mesure de la température réelle en tout point d'un circuit électronique en fonctionnement, le procédé d'application comportant trois étapes:

- une première étape de détermination de la matrice des émissivité à partir de la matrice de pixels correspondant à l'image de référence, la matrice des émissivité caractérisant l'émissivité de chaque point du circuit de référence représenté par un pixel sur l'image de référence;
- une seconde étape de recadrage durant laquelle on procède à la rotation de l'image de mesure, à la modification de son grandissement et la translation de cette image, de façon à construire une image de mesure corrigée dont les pixels coïncident avec ceux de l'image de référence;
- une troisième étape de correction durant laquelle la valeur de chaque pixels de la matrice de l'image infrarouge de mesure recadrée est modifiée en fonction de la valeur de l'émissivité mesurée à partir du pixel correspondant de l'image de référence.

Selon l'invention la deuxième étape du procédé d'application met en œuvre le procédé de recadrage selon l'invention.

Selon un mode de mise en œuvre préféré, on calcule durant la troisième étape l'énergie théoriquement émise par le circuit au point considéré. Le calcul est effectué à partir de la température mesurée pour ce point par la caméra infrarouge. On multiplie ensuite cette valeur par la valeur

de l'élément de la matrice d'émissivité correspondant, puis on calcul la valeur de la température réelle du circuit au point considéré en utilisant la valeur.

Le procédé selon l'invention permet avantageusement la mise en œuvre de mesures automatiques précises de températures à partir d'images infrarouges ce qui permet d'éviter la pose de capteurs sur le circuit lors du test en fonctionnement de ce dernier.

Les caractéristiques et avantages de l'invention seront mieux appréciés grâce à la description qui suit, description qui s'appuie sur les figures annexées qui représentent:

- la figure 1, un organigramme général du procédé selon l'invention
- la figure 2, une illustration présentant un exemple type d'image de référence;
- la figure 3, une illustration présentant un exemple type d'image de mesure, l'image de mesure présentant un décalage à l'image de référence;
- la figure 4, une illustration relative à la première sous-étape de la deuxième étape du procédé selon l'invention;
- les figures 5 à 8, des illustrations relatives à la deuxième sous-étape de la deuxième étape du procédé selon l'invention;
- les figures 9 à 14, des illustrations relatives à la troisième sous-étape de la deuxième étape du procédé selon l'invention;
- la figure 15 une illustration de l'image obtenue à l'issue de la deuxième sous-étape de la deuxième étape du procédé selon l'invention;
- la figure 16 une illustration de l'image infrarouge obtenue après recadrage et correction de la température au moyen des informations tirées de l'image de référence.
- la figure 17, un organigramme général d'un procédé de mesure de la température réelle d'un circuit à partir de son image infrarouge, ce procédé mettant en œuvre le procédé de recadrage d'image selon l'invention.

On décrit dans un premier temps les étapes du procédé selon l'invention tel qu'illustré par la figure 1.

Comme il a été dit précédemment, le procédé selon l'invention a en particulier pour objet d'effectuer le recadrage géométrique de l'image infrarouge d'un circuit sous test (image de mesure) et l'image infrarouge d'un circuit pris comme référence (image de référence), cette image étant réalisée alors que le circuit de référence est porté hors fonctionnement, à une température constante et uniforme.

L'image infrarouge du circuit de référence est exploitée sous la forme d'une matrice de pixels 12 chaque pixel étant repéré par sa position (x,y) sur l'image. Chaque pixel représente ici la valeur de la température au point de l'image considéré. Cette matrice de pixels est appelée matrice de référence.

L'image de mesure est ici définie par sa matrice de pixels 11 de manière identique à l'image de référence. Le recadrage géométrique a pour objet de permettre l'identification de zones homogènes de l'image de mesure aux zones homogènes correspondantes de l'image de référence et de permettre l'association automatique de la valeur d'émissivité déterminée pour chacune des zones homogènes de l'image de référence avec la zone correspondante de l'image de mesure.

Les zones homogènes étant définies aussi bien pour l'image de mesure que pour l'image de référence par les pixels qui les composent, il est nécessaire pour que l'association automatique d'une zone de l'image de mesure avec la valeur de l'émissivité correspondante tirée de l'image de référence soit possible, que l'orientation des deux images par rapport à la caméra infrarouge soit identique et que les deux images soient à la même échelle. Or, considérant que l'image de référence et l'image de mesure ont été réalisées dans des environnements différents, enceinte climatique pour l'image de référence et environnement de test pour l'image de mesure, il importe ici de considérer que les prises de vues correspondantes peuvent avoir été effectuées, comme l'illustrent les figures 2 et 3, avec un cadrage différent d'une prise de vue à l'autre.

Cette différence de cadrage a pour conséquence que les deux images ne peuvent pas être superposées l'une sur l'autre, de sorte qu'associer sans précaution à chaque pixel de la matrice 11 correspondant à l'image de

mesure, l'émissivité déterminée pour le pixel repéré par les mêmes coordonnées dans la matrice 12 de référence peut conduire, à associer à un pixel de l'image de mesure une valeur d'émissivité correspondant à un point du circuit représenté en réalité par un autre pixel de cette même image. Par suite la correction effectuée est erronée. Le rôle de la seconde étape est de réaliser sur l'image de mesure les opérations nécessaires au recadrage des deux images.

Comme l'illustre la figure 1, le procédé selon l'invention comporte trois étapes:

- une première étape 13 durant laquelle on effectue une rotation de l'image de mesure relativement à l'image de référence;
- une deuxième étape 14 durant laquelle on effectue une mise à l'échelle de l'image de mesure modifiée, obtenue à l'issue de la première sous-étape, relativement à l'image de référence;
- une troisième étape 15 durant laquelle on effectue une translation de l'image de mesure modifiée, obtenue à l'issue de la deuxième sous-étape, relativement à l'image de référence.

A l'issue de ces trois étapes on dispose d'une image infrarouge de mesure 16 recadrée par rapport à l'image de référence.

La première étape 13 met en œuvre trois opérations successives Ces opérations sont effectuées sur l'image de mesures ainsi que sur l'image de référence:

- une première opération, Illustrée par la figure 4, de détection des contours des zones apparaissant sur l'image infrarouge comme homogènes en température;
- une deuxième opération, illustrée par les figures 5 et 6, ayant pour objet de déterminer, sur l'image de mesure et sur l'image de référence, l'orientation majoritaire des lignes de contours ainsi définies par rapport à une direction absolue donnée;
- une troisième opération illustrée par les figures 6 et 8, ayant pour objet de comparer les orientations majoritaires respectives des images de mesure et de référence et de déterminer la rotation à appliquer à l'image de

mesure pour faire coïncider son orientation avec celle de l'image de référence.

Selon l'invention, les contours de chacune des images sont extraits par toute méthode connue. Dans un mode de mise en œuvre préféré on utilise la méthode connue sous le nom de "méthode de Canny". On rappelle ici que cette méthode commence par un filtrage gaussien de l'image considérée, qui permet de s'affranchir des problèmes liés au bruit de l'image. Elle se poursuit par le calcul et la sommation des gradients directionnels (verticaux et horizontaux) de façon à obtenir l' « intensité » du contour en chaque point. On ne conserve tout d'abord que les maxima locaux. La sélection des contours se fait ensuite par application d'une fonction de seuil à hystérésis aux maxima locaux ainsi déterminés. On détermine ainsi dans un premier temps si un point considéré fait ou non partie d'un contour. Ensuite, dans un second temps, on sélectionne les points contigus à un point du contour. On détermine ainsi, comme l'illustre la figure 4, une image 41 constituée des contours 42 des zones homogènes.

Selon l'invention encore, l'image des contours, est utilisée pour déterminer les directions de contours prépondérantes. La détermination peut, ici encore, être réalisée par toute méthode connue. Cependant, dans un mode de mise en œuvre préféré du procédé selon l'invention, on applique une transformation de Hough à l'image des contours. Cette transformation, connue par ailleurs permet de passer d'une représentation de l'image des contours dans un espace où chaque point représente un pixel (chaque pixel, représenté par ses coordonnées x et y est codé 0 ou 1 pour une image contour) à une représentation dans un espace où chaque point représente une droite repérée par sa distance au centre de l'image (ordonnée) et par l'angle qu'elle présente par rapport à une direction donnée de l'image, l'axe Oy par exemple (abscisse). Les figures 5 et 6 représentent respectivement le résultat obtenu par transformation de l'image de contours correspondant à l'image infrarouge de mesure 31 illustrée par la figure 3 et par transformation de l'image de contours correspondant à l'image infrarouge de référence 21 illustrée par la figure 2

Dans ce type de représentation, chaque droite du plan de l'image des contours est ainsi représentée par un pixel unique dans le plan de la transformée de Hough. La valeur de ce pixel est par ailleurs obtenu par un

accumulateur qui somme les valeurs (0 ou 1) des pixels de l'image correspondant à la droite considérée. Les droites pertinentes (qui passent par un grand nombre de points du contour) sont donc représentées par des pixels 51, 52, 61 ou 62 de forte intensité. Ainsi en sélectionnant les pixels de plus forte intensité on est à même de déterminer, pour l'image de mesure comme pour l'image de référence, les droites prépondérantes et leurs directions par rapport à une direction de référence donnée.

En pratique, s'agissant de l'analyse d'un circuit comportant une pluralité de composants il est raisonnable de considérer que les composants principaux sont de forme rectangulaire de sorte que les zones homogènes en température présentent également des contours rectilignes orthogonaux. De la sorte les droites prépondérantes apparaissent préférentiellement orientées perpendiculairement les unes par rapport aux autres. La représentation de Hough laisse donc apparaître une répartition des pixels de forte intensité le long de deux droites 53 et 54 ou 63 et 64 parallèles à l'axe des distances et séparées l'une de l'autre d'un écart sensiblement égal à 90°.

Selon l'invention enfin, la transformée de Hough de l'image des contours associée à l'image de mesure et la transformée de Hough de l'image des contours associée à l'image de référence sont ensuite exploitées conjointement pour calculer l'angle de rotation à appliquer à l'image de mesure pour que celle-ci puisse être superposé sur l'image de référence. Pour ce faire, on exploite les directions 53, 54 et 63, 64 selon lesquelles les contours identifiés sont majoritairement orientés.

En pratique, pour chaque valeur d'angle, on calcule, comme l'illustre la figure 7, le maximum de la transformée de Hough selon l'axe des distances au centre, de façon à obtenir un vecteur (Direction, intensité). L'opération est réalisée pour l'image de mesure et pour l'image de référence. On détermine ainsi les courbes de variation 71, 72 des intensités des vecteurs en fonction de l'orientation auxquelles on applique une corrélation circulaire, comme l'illustre la figure 8. Par suite, la valeur 82 d'angle de rotation pour laquelle on obtient un pic de corrélation 81 d'amplitude maximale, détermine le décalage angulaire α entre l'image de mesure et l'image de référence.

Ainsi, à l'issue de la première étape 13, on connaît la valeur de l'angle de rotation à applique à l'image de mesure pour lui donner une orientation identique à l'image de test.

La deuxième étape 14 met quant à elle en œuvre deux opérations successives:

- une première opération, illustrée par la figure 9 visant à déterminer des points d'intérêt sur l'image de mesure et sur l'image de référence;
- une seconde opération, illustrée par les figures 10 à 13, visant à déterminer le facteur d'échelle K liant l'image de mesure à l'image de référence, à partir des points d'intérêt définis précédemment.

En ce qui concerne la première opération de détermination des points d'intérêt, le procédé selon l'invention procède à la détection des coins présents sur l'image. On rappelle ici qu'un coin peut être défini comme l'intersection de deux contours. Un coin peut aussi être défini comme un point pour lequel il y a deux directions de contours prépondérantes différentes dans un voisinage local du point. Plus généralement, dans une image, un point d'intérêt peut être défini comme un point ayant une position bien définie. Par suite, un coin constitue un point d'intérêt, qui peut être reconnu d'une image donnée d'un objet, une image infrarouge dans le cas présent, à une autre image du même objet.

Il existe plusieurs méthodes connues de l'homme du métier permettant la détection des coins présents sur une image. Cependant, dans un mode de mise en œuvre préféré, le procédé selon l'invention met en œuvre la méthode connue sous le nom de "méthode de Harris".

Conformément à la méthode de Harris, on détermine les gradients G_x et G_y de l'image considérée. Les matrices $A=(G_x)^2$, $B=(G_y)^2$ et $A=(G_x)^2$ sont ensuite calculées, puis on applique un filtrage gaussien 2D à ces matrices. On calcule ensuite l'opérateur R défini par la relation $R=(A\times B-C^2)-k\times(A+B)$, où k est un facteur empirique arbitraire dont la valeur typique est 0,04. Puis on retient les coins les plus significatifs à 5 pixels près.

Le résultat de cette première opération peut être représenté comme l'illustre la figure 9 par une image sur laquelle sont représentés les différents coins 91 ainsi déterminés. L'objet de la suite de l'étape 14 est de déterminer parmi ces coins, ceux qui correspondent réellement aux contours d'une zone.

En ce qui concerne la seconde opération, le calcul du facteur d'échelle proprement dit, celle-ci consiste dans un premier temps à définir des sous-images de l'image considérée, chaque sous-image, d'une taille de quelques pixels carrés, étant centrée sur un coin 91. La taille de chaque sous-image est par ailleurs fixée de manière à obtenir un résultat satisfaisant, par exemple à 5 pixels (verticalement et horizontalement) autour du point d'intérêt considéré. Le but de cette opération est de limiter le nombre de points d'intérêt à partir desquels lesquels le calcul du facteur d'échelle est effectué, en ne conservant que les points d'intérêt (i. e. les coins) correspondant réellement au contour d'une zone homogène. Cette opération de segmentation en sous-image est réalisée à la fois sur l'image de mesure et sur l'image de référence.

On détermine ensuite pour chaque sous-image les moments de Hu (ou descripteurs de Hu) qui lui sont associés. Les descripteurs de Hu se présentent avantageusement comme des invariants algébriques qui peuvent être calculés sur des images. Ces invariants restent identiques pour les différentes similitudes (ils sont en revanche sensibles à l'illumination). On rappelle ici que pour une image donnée les moments de Hu sont définis à partir à partir des moments normalisés μ_{p+q} d'ordre 3 de l'image par les relations suivantes:

$$
\begin{aligned}
\phi_1 &= \mu_{2,0} + \mu_{0,2} \\
\phi_2 &= (\mu_{2,0} - \mu_{0,2})^2 + 4\mu_{1,1}^2 \\
\phi_3 &= (\mu_{3,0} - 3\mu_{1,2})^2 + (3\mu_{2,1} - \mu_{0,3})^2 \\
\phi_4 &= (\mu_{3,0} + \mu_{1,2})^2 + (\mu_{2,1} + \mu_{0,3})^2 \\
\phi_5 &= (\mu_{3,0} - 3\mu_{1,2}) \cdot (\mu_{3,0} + \mu_{1,2}) \cdot \left[(\mu_{3,0} + \mu_{1,2})^2 - 3(\mu_{2,1} + \mu_{0,3})^2\right] \\
&\quad + (3\mu_{2,1} - \mu_{0,3}) \cdot (\mu_{2,1} + \mu_{0,3}) \cdot \left[3(\mu_{3,0} + \mu_{1,2})^2 - (\mu_{2,1} + \mu_{0,3})^2\right] \\
\phi_6 &= (\mu_{2,0} - \mu_{0,2}) \cdot \left[(\mu_{3,0} + \mu_{1,2})^2 - (\mu_{2,1} + \mu_{0,3})^2\right] \\
&\quad - 4\mu_{1,1} \cdot (\mu_{3,0} + \mu_{1,2}) \cdot (\mu_{2,1} + \mu_{0,3}) \\
\phi_7 &= (3\mu_{2,1} - \mu_{0,3}) \cdot (\mu_{3,0} + \mu_{1,2}) \cdot \left[(\mu_{3,0} + \mu_{1,2})^2 - 3(\mu_{2,1} + \mu_{0,3})^2\right] \\
&\quad + (\mu_{3,0} - 3\mu_{1,2}) \cdot (\mu_{2,1} + \mu_{0,3}) \cdot \left[3(\mu_{3,0} + \mu_{1,2})^2 - (\mu_{2,1} + \mu_{0,3})^2\right]
\end{aligned}
\qquad [1]
$$

Les moments d'ordres $p+q$ d'une image sont par ailleurs définis par la formule :

$$\nu_{p,q} = \int_{R^2} x^p \cdot y^q \cdot I(x, y) \cdot dx, dy \qquad [2]$$

où I(x,y) représente la valeur du pixel situé à la position (x,y) sur l'image dans un repère fixe par exemple lié à la caméra.

Chaque sous-image de l'image de mesure et de l'image de référence est alors ainsi représentée par un vecteur contenant les sept moments de Hu: $(\phi_1,\phi_2,\phi_3,\phi_4,\phi_5,\phi_6,\phi_7)$. Une matrice de distance D est alors calculée entre les sous-images constituant les deux images, l'élément situé à la i-ème ligne et la j-ème colonne de la matrice étant défini en utilisant la relation suivante:

$$D_{i,j} = \sqrt{\sum_{m=1}^{7} \left(\phi_m(i) - \phi_m(j)\right)^2}$$

où i représente la i-ème sous-image de l'image de mesure et j la j-ème sous-image de l'image de référence.

Selon l'invention, les éléments de cette matrice font ensuite l'objet d'un appariement d'une image à l'autre. Une sous-image n de l'image de mesure est appariée à une sous-image m de l'image de référence si pour cette sous-image n, la distance $D_{n,m}$ représente la valeur minimum de $D_{n,j}$ et si pour la sous-image m la distance $D_{n,m}$ représente la valeur minimum de $D_{i,m}$. L'appariement s'effectue ainsi en deux analyse simultanées. Autrement dit, la sous-image i de l'image 1 n'est apparié à la sous-image j de l'image 2 que si i reconnaît j comme son image et que réciproquement j reconnaît i comme son image. Par suite le processus d'appariement est un processus itératif pour lequel après chaque itération les appariements effectués sont retirés des appariements possibles, une image ne pouvant être appariée qu'à une seule image. A l'issue de ce processus on peut définir une matrice de distance dite "réduite" qui, comme l'illustre la représentation de la figure 10

contient, sous-forme réduite, pour chaque point 101 représentant une distance séparant deux sous-image appariées suivant deux axes orthogonaux de référence Ox, Oy, le nombre de sous-images appariées correspondant à ce point.

L'opération d'appariement ayant été réalisée, les images appariées peuvent alors être déduites l'une de l'autre, à condition que la rotation ait été effectuée correctement, par simple application d'une homothétie et/ou d'une translation. La suite de la description expose une façon de déterminer le rapport d'homothétie K.

Soient X_i et Y_i les coordonnées de la sous-image i dans l'image de mesure et X_j, Y_j de la sous-image j dans l'image de référence. Les sous-images i et j étant appariées, on peut écrire les relations suivantes:

$$X_i = K \times X_j + X_0,$$

$$Y_i = K \times Y_j + Y_0$$

$$X_i - Y_i = K \times (X_j - Y_j) + (X_0 - Y_0).$$

Où K représente le facteur d'échelle entre les 2 images, X_0 le décalage, exprimé en nombre de pixel, selon l'axe horizontal entre les 2 images et Y_0 le décalage vertical.

Chaque appariement de deux sous-images correspond donc comme l'illustrent les figures 11 à 13, à une droite dans les espaces respectifs (K, X_0), (K, Y_0) et $(K, X_0 - Y_0)$ de sorte que, dans la pratique, en fonction des appariements retenus, les droites correspondant aux différents appariements effectués se coupent majoritairement en un point 111, 121, et 131 dont l'abscisse correspond au facteur d'échelle K qui définit le grandissement relatif des deux images et dont l'ordonnée correspond selon la représentation choisie à un décalage horizontal (figure 12) ou vertical (figure 13) ou bien un décalage dans un plan (figure 11). Il est donc possible de déterminer, selon la qualité de l'appariement effectué au préalable, un ou plusieurs points au niveau desquels un nombre maximum de droites se coupent, chaque point définissant un facteur d'échelle K particulier. Par suite, selon l'invention, les maximums de chaque plan pour chaque facteur

d'échelle K sont alors sommés pour déterminer le facteur d'échelle 141 le plus probable comme l'illustre la figure 14. On détermine ainsi avantageusement le facteur d'échelle à appliquer pour passer de l'image de référence à l'image de mesure et inversement.

La Troisième étape 15 est appliquée à l'image de mesure corrigée 151 illustrée par la figure 15. Cette image corrigée, réorientée et remise à l'échelle par rapport à l'image de référence ne présente plus, par rapport à cette image, dans le cas le plus défavorable, qu'un décalage en translation matérialisé par le décalage des systèmes d'axes 152 et 153 sur la figure 15. Par suite la troisième sous-étape 23 consiste essentiellement à opérer une translation de l'image de mesure modifiée, de façon à superposer les deux systèmes d'axes. Pour ce faire, une corrélation croisée suivant les deux dimensions est appliquée à l'image de mesure de façon à déterminer les déplacements horizontaux et verticaux qui permettent de maximiser la similitude des 2 images. Une fois déterminés, ces déplacements sont appliqués à l'image corrigée 151 de sorte que l'on obtient alors l'image recadrée 16 présentée sur la figure 16.

A l'issue de la troisième étape 15 du procédé selon l'invention on dispose ainsi d'une image de référence et d'une image de mesure 16 (cf. figure 17) cadrée de la même façon que l'image de référence (cf. figure 3).

Il est possible d'utiliser le procédé selon l'invention décrit précédemment dans une application générale consistant en la détermination automatique des points chauds sur un circuit électrique ou électronique mis en test de bon fonctionnement, après fabrication par exemple.

Dans cette application, on exploite simultanément l'image infrarouge du circuit sous test (image de mesure) et l'image infrarouge du circuit pris comme référence (image de référence), cette image étant réalisée alors que le circuit de référence est porté hors fonctionnement, à une température constante et uniforme.

Le procédé pour effectuer une telle détermination comporte notamment, comme l'illustre la figure 17 les étapes suivantes:

- une première étape 171 de calcul de l'émissivité en chaque point du circuit considéré;
- une deuxième étape 172 de recadrage de l'image infrarouge réalisée une image de référence;
- une troisième étape 173 de correction durant laquelle la valeur de chaque pixels de la matrice de l'image infrarouge de mesure recadrée est modifiée en fonction de la valeur de l'émissivité mesurée à partir du pixel correspondant de l'image de référence.

La première étape 171 consiste tout d'abord à déterminer à partir d'une image de référence la matrice d'émissivité correspondante. L'image infrarouge du circuit de référence est exploitée sous la forme d'une matrice de pixels 12 chaque pixel étant repéré par sa position (x,y) sur l'image. Chaque pixel représente ici la valeur de la température au point de l'image considéré. Cette matrice de pixels est appelée matrice de référence.

La première étape 171 consiste ensuite à comparer, pour chacun des pixels de l'image 12, la valeur de température associée à ce pixel avec la température réelle du point correspondant sur le circuit de référence celui-ci étant porté à une température constante et uniforme. La différence entre la température réelle et la température mesurée est ensuite utilisée pour déterminer l'émissivité du circuit au point. On rappelle ici que l'émissivité ε d'un élément correspond au rapport de l'énergie thermique émise (rayonnée) par l'élément considéré pour une température donnée à l'énergie émise (rayonnée) par un corps noir porté à la même température (rapport sans dimension inférieur à 1). L'énergie thermique est ici déterminée à partir de la Loi de Planck et de la réponse du capteur de la caméra infrarouge en fonction des longueurs d'onde reçues. On obtient ainsi à l'issue de la première étape une matrice d'émissivités qui caractérise chaque point du circuit de référence par son émissivité.

La deuxième étape 172 met en œuvre les trois étapes 13, 14 et 15 du procédé de recadrage selon l'invention décrit précédemment. Elle n'est donc pas détaillée ici.

La troisième étape 173 consiste quant à elle à calculer la température réelle du circuit en chaque point en utilisant la température déterminée à partir de l'image de mesure recadrée, obtenue à l'issue de l'étape 172, et la valeur de l'émissivité du point considéré. On produit ainsi une matrice de mesures corrigées 16, qui permet de déterminer les écarts réels de température entre les différentes zones du circuit et d'identifier les zones où la température peut être considérée comme révélatrice d'un problème de disfonctionnement. Par suite, à partir de la valeur du pixel correspondant de l'image corrigée, on obtient de manière automatique la valeur réelle de la température du point du circuit représenté par ce pixel.

De manière classique, la température déduite de l'image infrarouge de mesure est corrigée en appliquant pour chaque pixel de la matrice de mesure 11, ou pour chaque zone homogène en température, la méthode suivante:

Soit f la fonction reliant la température du point considéré à l'énergie vue par la caméra pour le pixel correspondant à ce point pour une émissivité parfaite.

À partir de l'image de calibration, on peut donc déterminer une image de l'énergie vue par la caméra. Or, la température de calibration est connue et l'on peut aussi déterminer l'énergie théorique qui aurait dû être vue pour une émissivité parfaite (égale à 1). La matrice des émissivités est alors obtenue comme le rapport entre l'énergie réellement vue et l'énergie théorique attendue.

À partir de l'image de mesure, on peut également déterminer la matrice des énergies vues. En divisant cette énergie vue pour chaque pixel par l'émissivité correspondante, on obtient l'énergie qui aurait été émise si l'émissivité avait été parfaite. On peut alors utiliser la fonction inverse de f pour déterminer la température réelle.

REVENDICATIONS

1. Procédé pour réaliser en automatique le recadrage d'une image de mesure infrarouge (31) d'un circuit en fonctionnement sur une image infrarouge de référence (21) réalisée sur un circuit semblable hors fonctionnement et porté à une température uniforme, les deux images infrarouges étant exploitées sous la forme de matrices de pixels (11, 12), caractérisé en ce qu'il comporte les étapes suivantes:

- une première étape (13) durant laquelle on effectue une rotation de l'image de mesure (31), la valeur de l'angle de rotation étant déterminée par détection des contours (42) des zones de l'image homogènes en température, par détermination de directions prépondérantes dans l'image (31) et par détermination de l'écart angulaire séparant ces directions prépondérantes des directions prépondérantes correspondantes de l'image de référence (21);

- une deuxième étape (14) durant laquelle on modifie la taille de l'image de mesure réorientée obtenue à l'issue de l'étape précédente (13), la taille de l'image étant modifiée d'un facteur d'échelle K, le facteur K étant déterminé par appariement des points caractéristiques des deux images, et analyse de l'écart entre la distance des points caractéristiques de l'image de mesure corrigée à une origine donnée et la distance à cette même origine des points caractéristiques de l'image de référence auxquels ils sont appariés;

- une troisième étape (15) durant laquelle l'image (151) obtenue à l'issue de la deuxième étape (14) est translatée de façon à superposer le repère de position (152) de cette image au repère de position (153) de l'image de référence.

2. Procédé selon la revendication 1, caractérisé en ce que durant la première étape (13) on effectue la détection des contours de l'image de mesure (31) et de l'image de référence (21) par la méthode de Canny.

3. Procédé selon la revendication 1 ou 2, caractérisé en ce que durant la première étape (13) on détermine le décalage angulaire des

droites prépondérantes en appliquant une transformation de Hough dans un plan (distance, angle de rotation) aux pixels constituant les contours (42) des images et en effectuant une corrélation circulaire entre les vecteurs des maxima de la transformée de hough de chacune des images.

4. Procédé selon l'une des revendications 1 à 3, caractérisé en ce que durant la deuxième étape (14) les points d'intérêt recherchés sont des coins (91) déterminés au moyen du détecteur de Harris.

5. Procédé selon la revendication 4, caractérisé en ce que durant la deuxième étape (14) la détermination du facteur d'échelle K est réalisée en découpant l'image de mesure corrigée en rotation et l'image de référence (21) en sous-images, chaque sous-image étant centrée sur un coin (91) et étant caractérisée par ses descripteurs de Hu puis en effectuant l'appariement des sous images de l'image de mesure corrigée en rotation avec les sous-images de l'image de référence (21), l'appariement étant réalisé en calculant les éléments $D_{i,j}$ de la matrice de distance D définie par:

$$D_{i,j} = \sqrt{\sum_{m=1}^{7} \left(\phi_m(i) - \phi_m(j)\right)^2}$$

où i représente la i-ème sous-image de l'image de mesure et j la j-ème sous-image de l'image de référence et $\phi_m(i)$ le moment de Hu d'indice m de la sous-image i.

6. Procédé selon l'une quelconque des revendications 1 à 5, caractérisé en ce que durant la troisième étape (15) on effectue une corrélation croisée suivant les deux dimensions de l'image de mesure corrigée (151) obtenue à l'issue de la deuxième étape (14) et de l'image de référence (21), de façon à déterminer par rapport à deux directions perpendiculaires l'écart de positionnement existant entre les deux images.

7. Utilisation du procédé selon l'une quelconque des revendications précédentes pour réaliser en automatique la mesure de la température réelle d'un circuit en fonctionnement à partir de son image infrarouge (31), caractérisée en ce que le recadrage de l'image infrarouge du circuit (31) sur l'image infrarouge de référence (21) est suivi d'une étape (173) de correction durant laquelle la valeur de chaque pixel de la matrice de l'image infrarouge de mesure (31) recadrée (16) est modifiée en fonction de la valeur de l'émissivité mesurée à partir du pixel correspondant de l'image de référence (21).

8. Utilisation selon la revendication 7, caractérisée en ce que durant l'étape de correction (173) on calcule l'énergie théoriquement émise par le circuit au point considéré, à partir de la température mesurée pour ce point par la caméra infrarouge, on multiplie cette valeur par la valeur de l'élément de la matrice d'émissivité correspondant, puis on calcul la valeur de la température réelle du circuit au point considéré en utilisant la valeur.

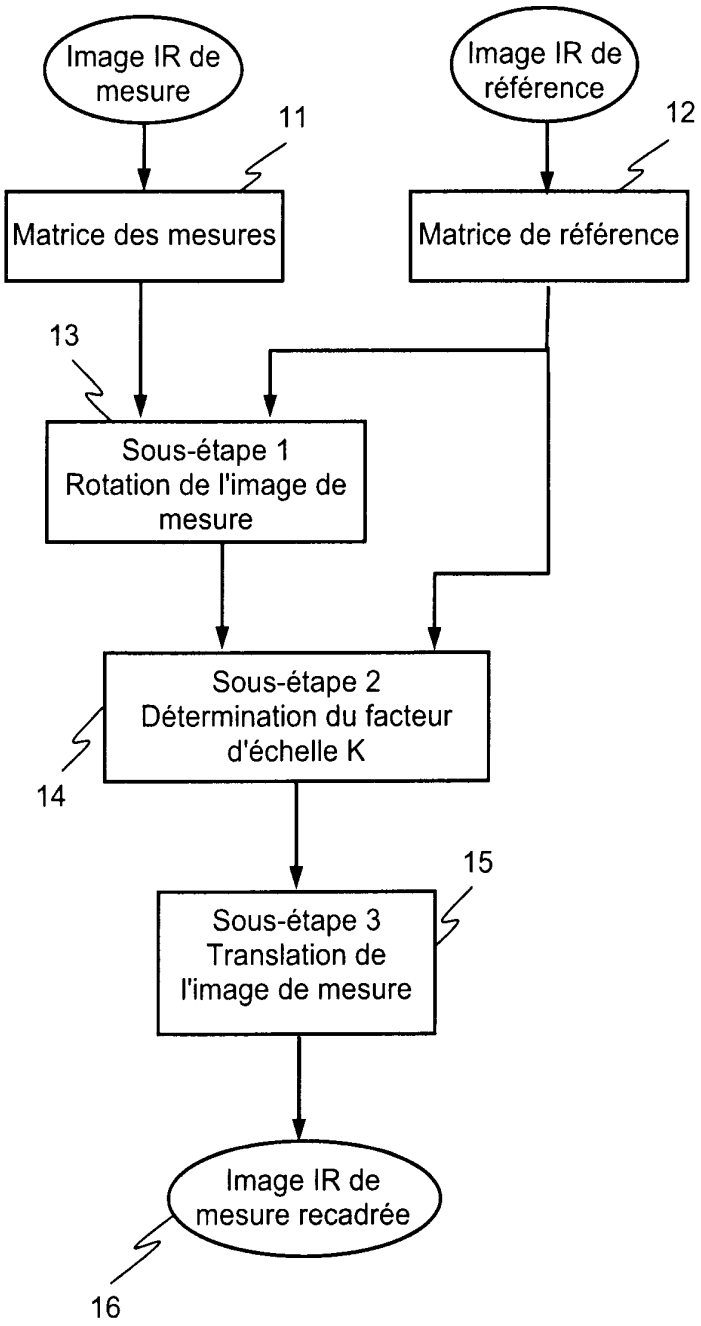


Image IR de mesure
Image IR de référence
11
12
Matrice des mesures
Matrice de référence
13
Sous-étape 1
Rotation de l'image de mesure
Sous-étape 2
Détermination du facteur d'échelle K
14
15
Sous-étape 3
Translation de l'image de mesure
Image IR de mesure recadrée
16

Fig. 1

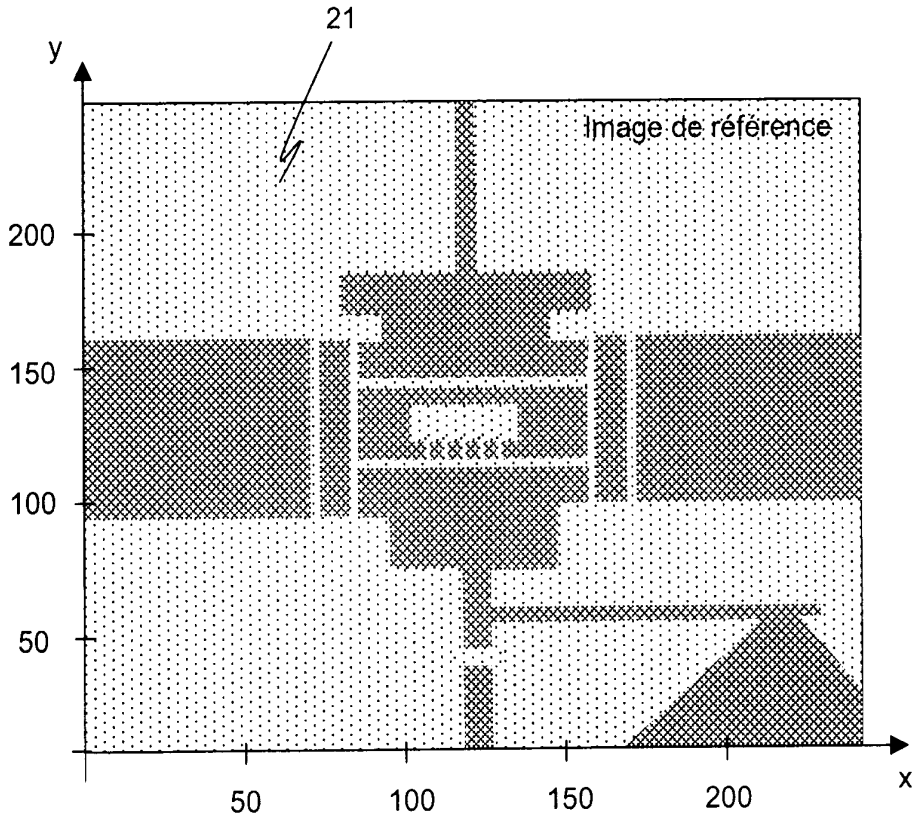


Fig. 2

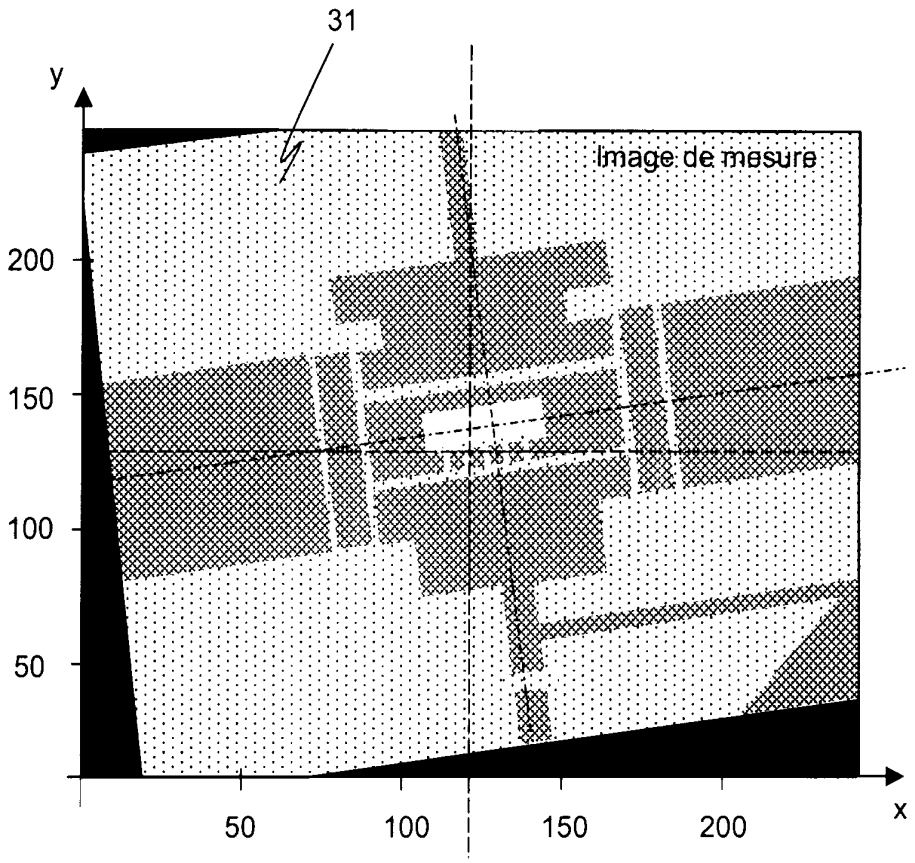


Fig. 3

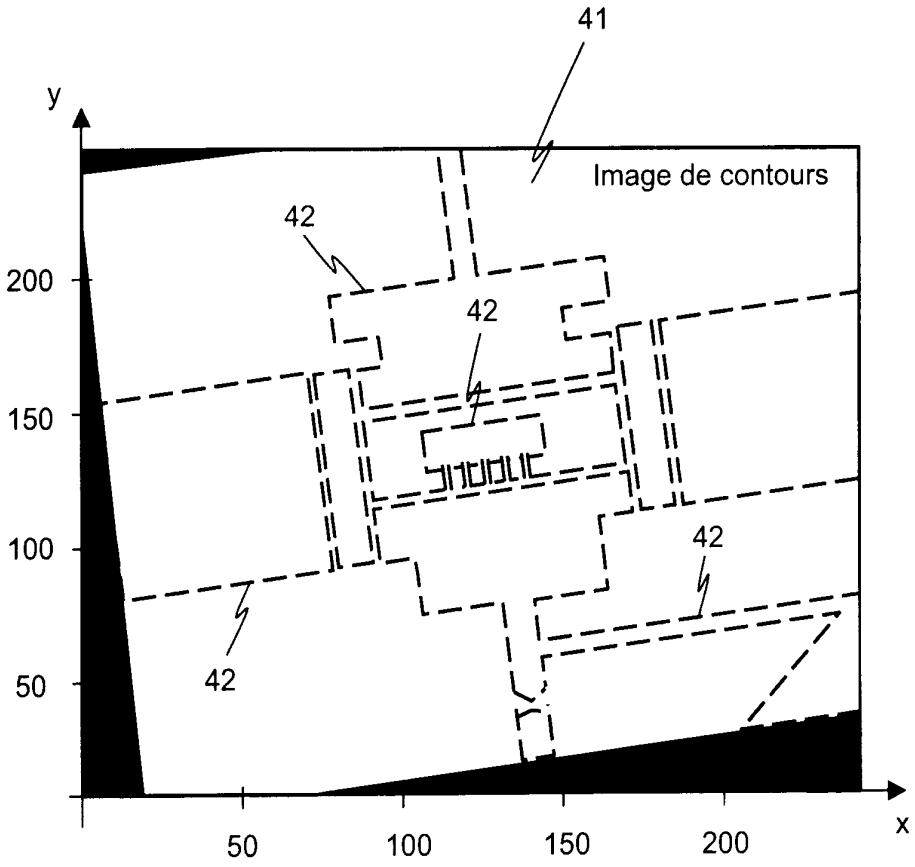


Fig. 4

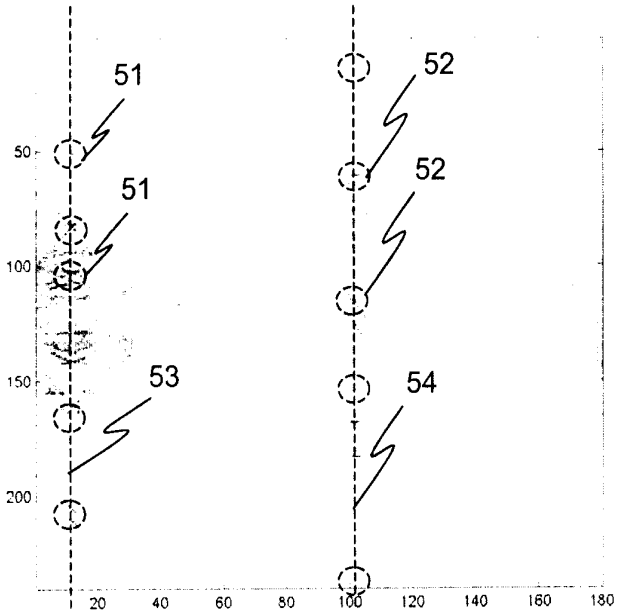

51
51
53
52
52
54
50
100
150
200
20
40
60
80
100
120
140
160
180

Fig. 5

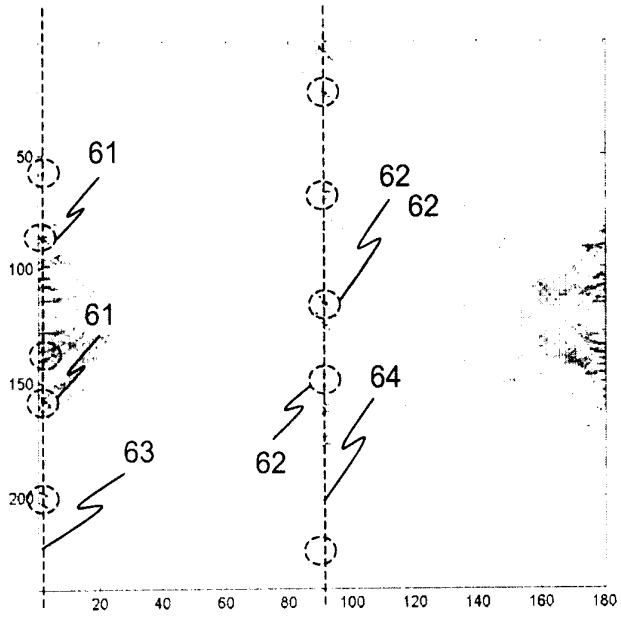

61
61
63
62
62
64
62
50
100
150
200
20
40
60
80
100
120
140
160
180

Fig. 6

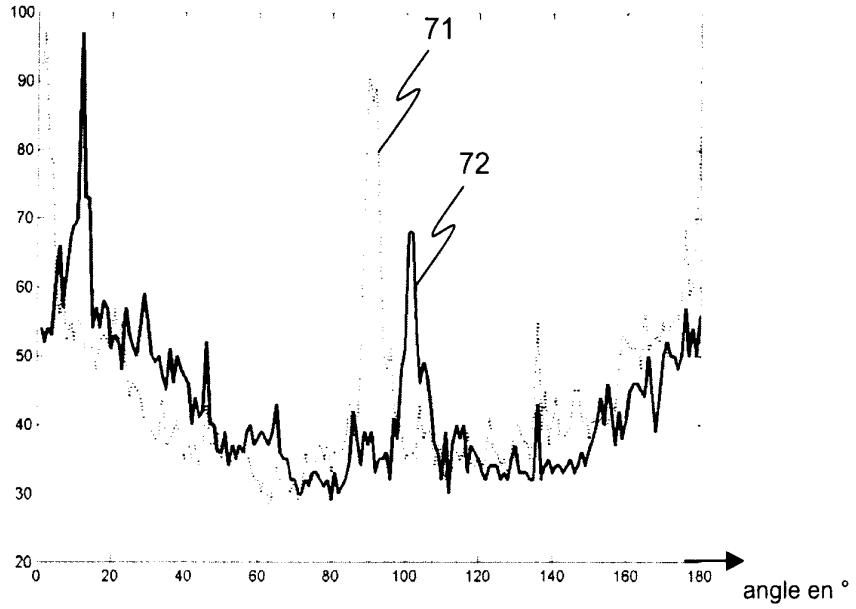

71
72
100
90
80
70
60
50
40
30
20
0
20
40
60
80
100
120
140
160
180
angle en °

Fig. 7

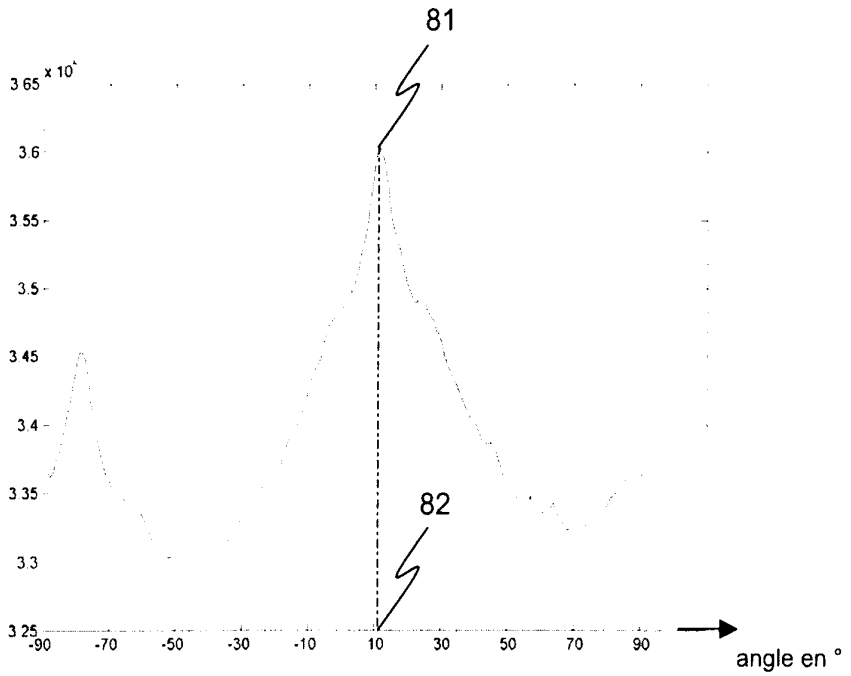

81
82
x 10^4
3.65
3.6
3.55
3.5
3.45
3.4
3.35
3.3
3.25
-90
-70
-50
-30
-10
10
30
50
70
90
angle en °

Fig. 8

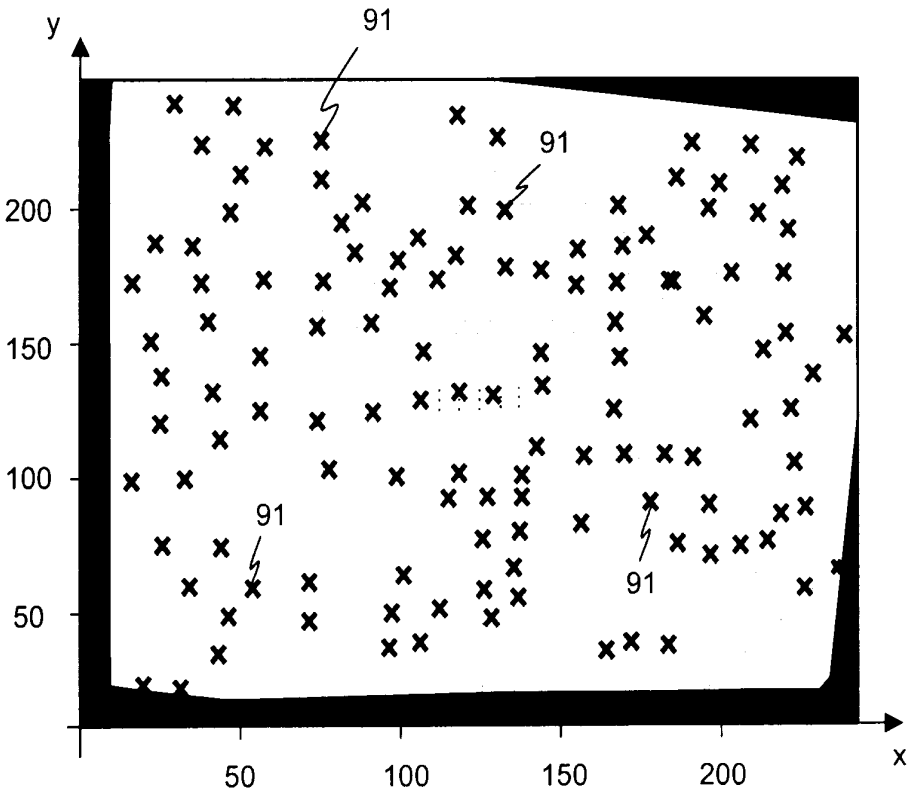


Fig. 9

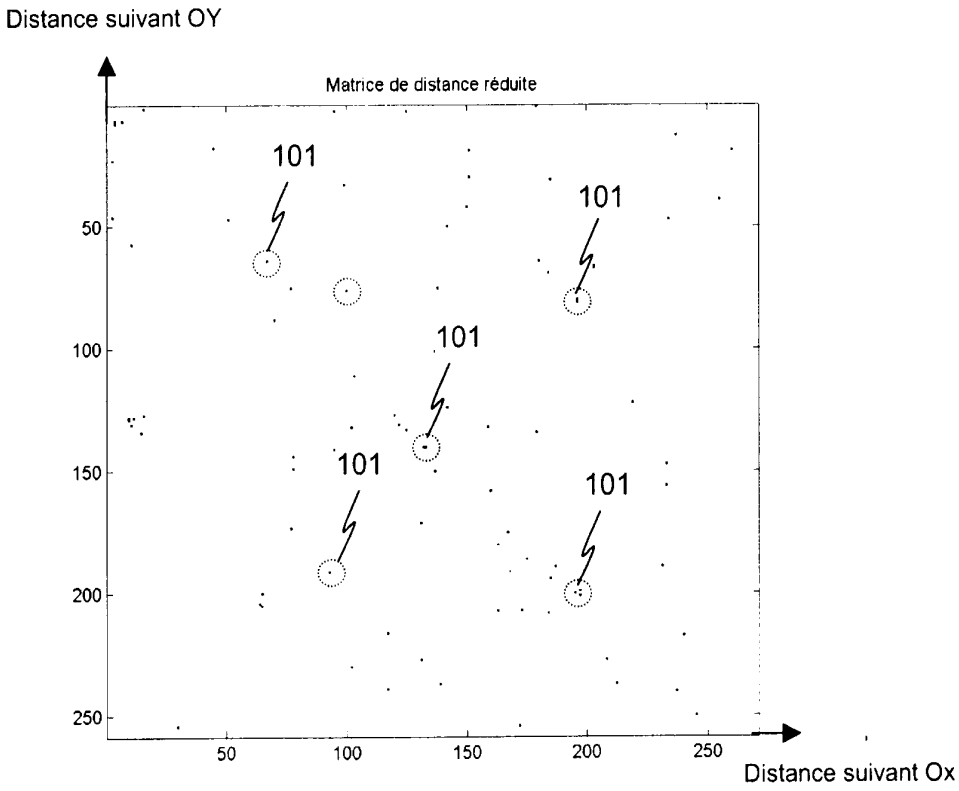

Distance suivant OY
Matrice de distance réduite
101
101
101
101
101
50
100
150
200
250
50
100
150
200
250
Distance suivant Ox

Fig. 10

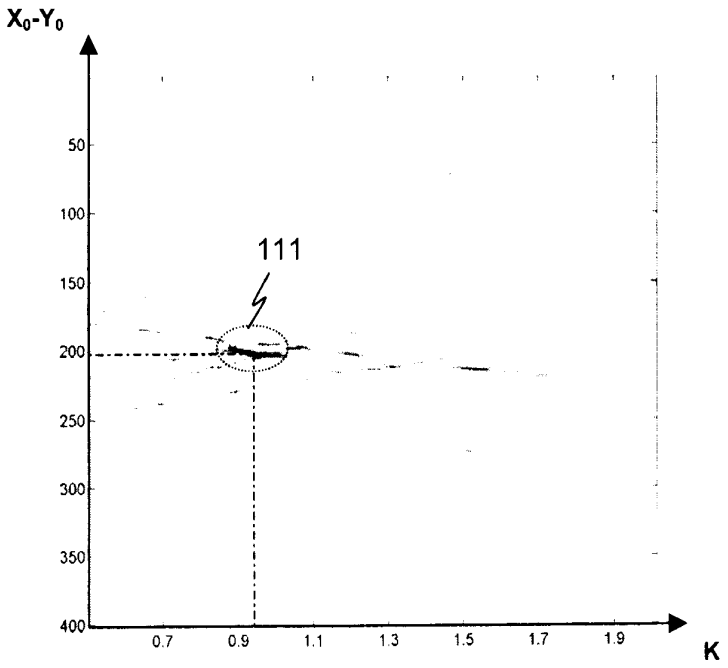

X0-Y0
50
100
150
200
250
300
350
400
111
0.7
0.9
1.1
1.3
1.5
1.7
1.9
K

Fig. 11

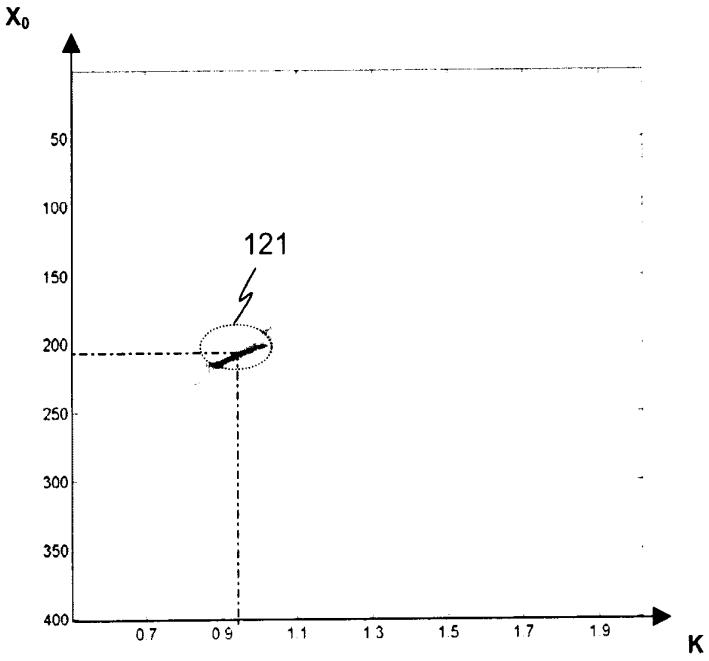

X0
50
100
150
200
250
300
350
400
121
0.7
0.9
1.1
1.3
1.5
1.7
1.9
K

Fig. 12

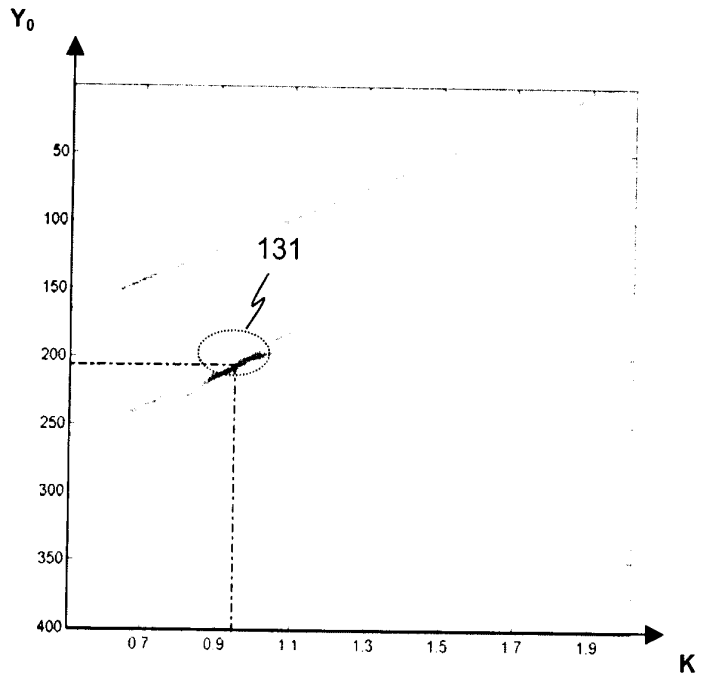

Y0
50
100
150
200
250
300
350
400
131
0 7
0 9
1 1
1 3
1 5
1 7
1 9
K

Fig. 13

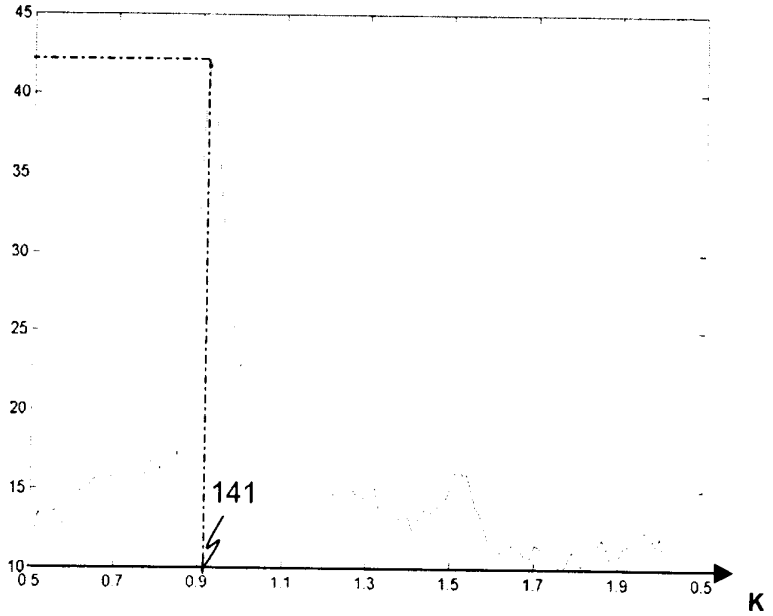

45
40
35
30
25
20
15
10
141
0 5
0 7
0 9
1 1
1 3
1 5
1 7
1 9
0 5
K

Fig. 14

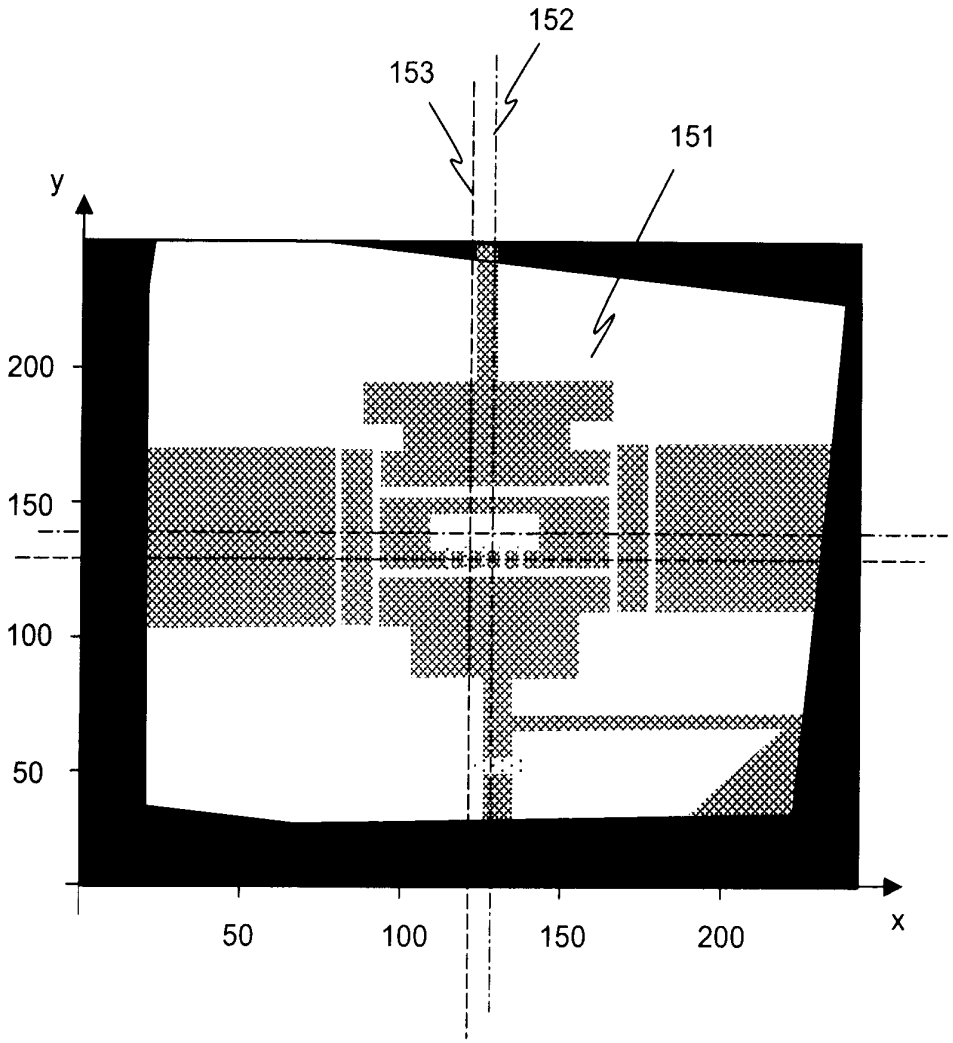


Fig. 15

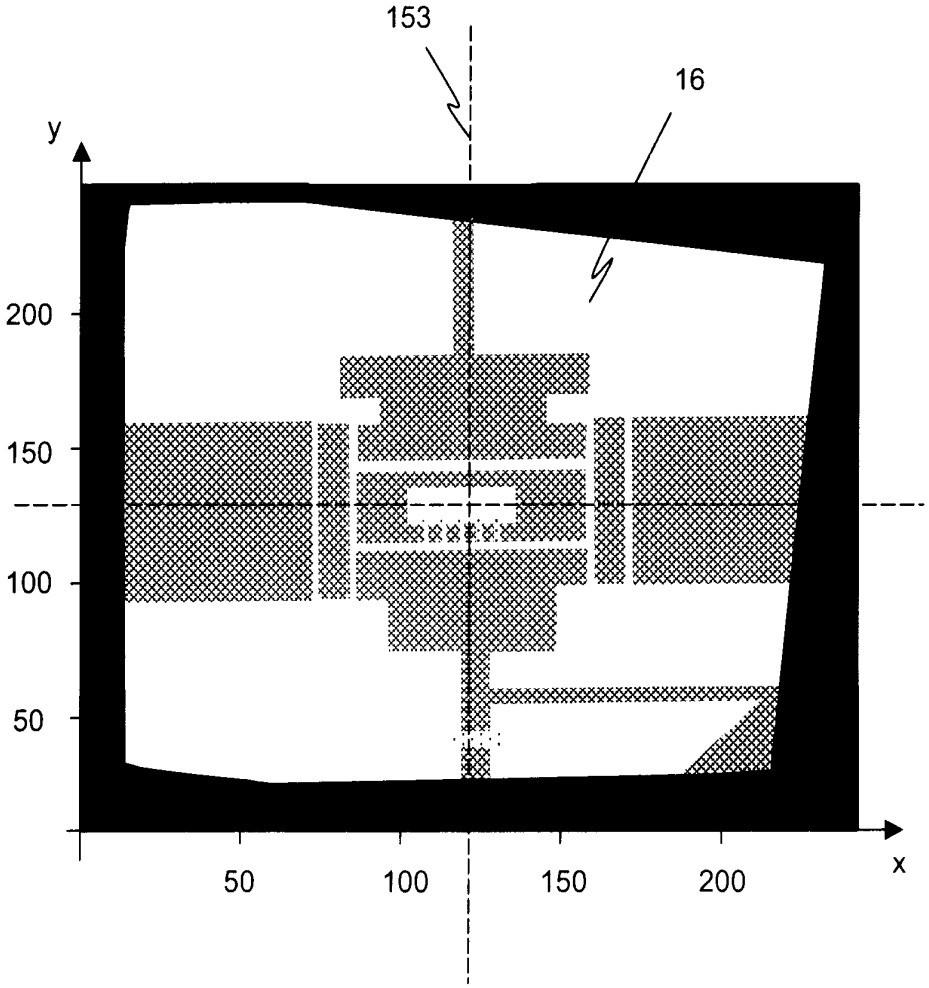


Fig. 16

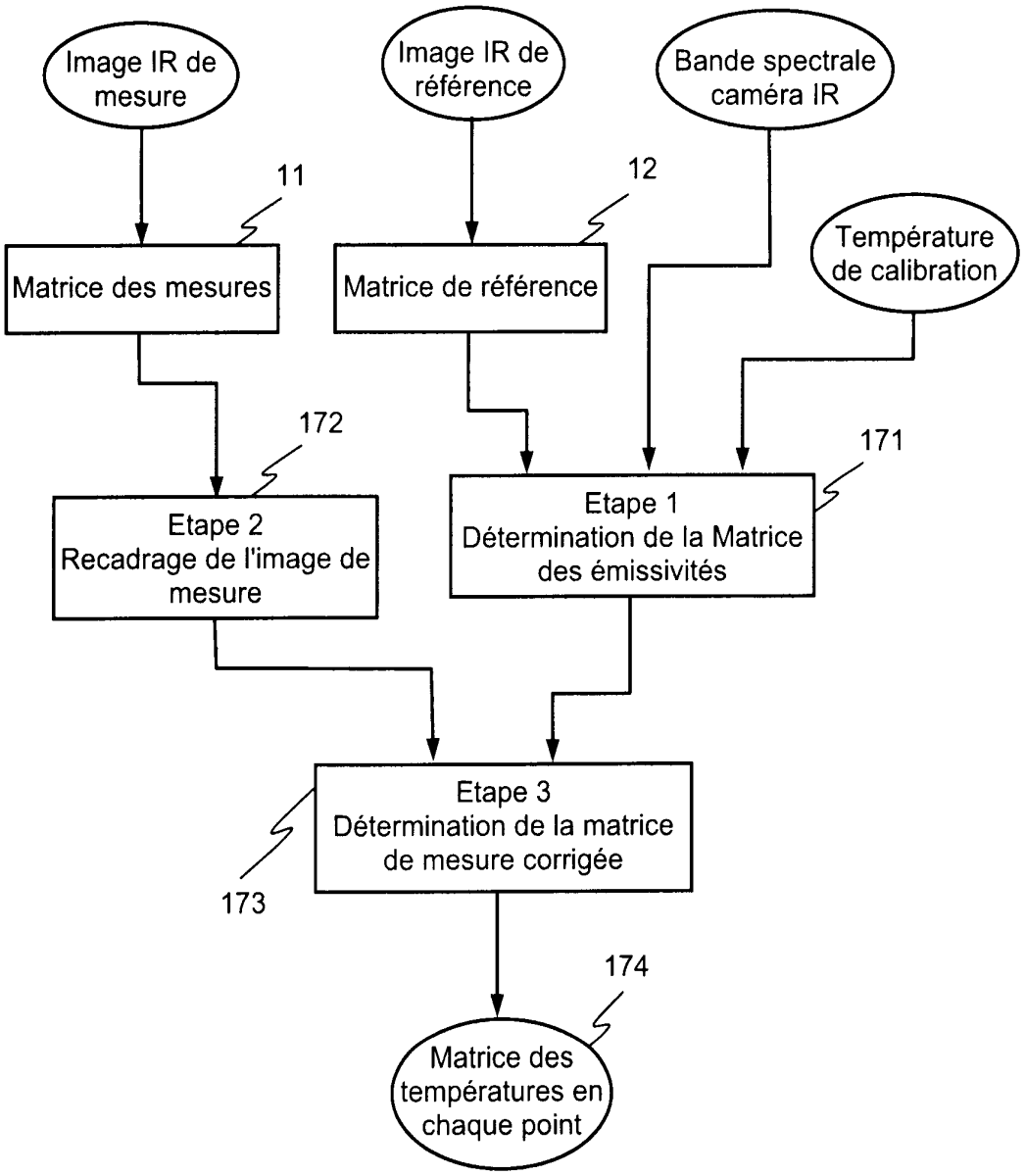


Image IR de mesure
Image IR de référence
Bande spectrale caméra IR
11
12
Température de calibration
Matrice des mesures
Matrice de référence
172
171
Etape 2
Recadrage de l'image de mesure
Etape 1
Détermination de la Matrice des émissivités
Etape 3
Détermination de la matrice de mesure corrigée
173
174
Matrice des températures en chaque point

Fig. 17

RÉPUBLIQUE FRANÇAISE

INSTITUT NATIONAL DE LA PROPRIÉTÉ INDUSTRIELLE

RAPPORT DE RECHERCHE PRÉLIMINAIRE

établi sur la base des dernières revendications déposées avant le commencement de la recherche

N° d'enregistrement national

FA 715697
FR 0806074

DOCUMENTS CONSIDÉRÉS COMME PERTINENTS		Revendication(s) concernée(s)	Classement attribué à l'invention par l'INPI
Catégorie	Citation du document avec indication, en cas de besoin, des parties pertinentes		
Y	ISTENIC R ET AL: "Thermal and Visual Image Registration in Hough Parameter Space" SYSTEMS, SIGNALS AND IMAGE PROCESSING, 2007 AND 6TH EURASIP CONFERENCE FOCUSED ON SPEECH AND IMAGE PROCESSING, MULTIMEDIA COMMUNICATIONS AND SERVICES. 14TH INTERNATIONAL WORKSHOP ON, IEEE, 1 juin 2007 (2007-06-01), pages 106-109, XP031159568 * le document en entier *	1-8	G06T3/00 H04N1/387 H04N5/33 G01K17/00 G01R31/308
Y	ZITOVA B ET AL: "IMAGE REGISTRATION METHODS: A SURVEY" IMAGE AND VISION COMPUTING, GUILDFORD, GB, vol. 21, no. 11, 1 octobre 2003 (2003-10-01), pages 977-1000, XP002522120 * page 990, colonne de droite, alinéa 2 *	1-8	
A	YIN Z ET AL: "Thermal and Visual Image Processing and Fusion" SIMTECH TECHNICAL REPORT, vol. TR0630, 1 janvier 2000 (2000-01-01), pages 1-6, XP009116836 * page 2, colonne de gauche, alinéa 3 - colonne de droite, alinéa 2 *	1-8	DOMAINES TECHNIQUES RECHERCHÉS (IPC) G06T
A	US 2006/276698 A1 (HALLDORSSON GISLI H [IS] ET AL) 7 décembre 2006 (2006-12-07) * alinéa [0053] - alinéa [0064] *	1-8	

-/--

Date d'achèvement de la recherche	Examinateur
26 août 2009	Pierfederici, A

CATÉGORIE DES DOCUMENTS CITÉS

X : particulièrement pertinent à lui seul
Y : particulièrement pertinent en combinaison avec un autre document de la même catégorie
A : arrière-plan technologique
O : divulgation non-écrite
P : document intercalaire

T : théorie ou principe à la base de l'invention
E : document de brevet bénéficiant d'une date antérieure à la date de dépôt et qui n'a été publié qu'à cette date de dépôt ou qu'à une date postérieure.
D : cité dans la demande
L : cité pour d'autres raisons

& : membre de la même famille, document correspondant

1

EPO FORM 1503 12.99 (P04C14)

RÉPUBLIQUE FRANÇAISE

INSTITUT NATIONAL
DE LA PROPRIÉTÉ
INDUSTRIELLE

RAPPORT DE RECHERCHE PRÉLIMINAIRE

établi sur la base des dernières revendications
déposées avant le commencement de la recherche

N° d'enregistrement national

FA 715697
FR 0806074

DOCUMENTS CONSIDÉRÉS COMME PERTINENTS		Revendication(s) concernée(s)	Classement attribué à l'invention par l'INPI
Catégorie	Citation du document avec indication, en cas de besoin, des parties pertinentes		
A	CORIAS E ET AL: "Segment-based registration technique for visual-infrared images" OPTICAL ENGINEERING, SOC. OF PHOTO-OPTICAL INSTRUMENTATION ENGINEERS, BELLINGHAM, vol. 39, no. 1, 1 janvier 2000 (2000-01-01), pages 282-289, XP009116835 * page 284, colonne de gauche, alinéa 2 - alinéa 9 * -----	1-8	
			DOMAINES TECHNIQUES RECHERCHÉS (IPC)

Date d'achèvement de la recherche	Examinateur
26 août 2009	Pierfederici, A

CATÉGORIE DES DOCUMENTS CITÉS

X : particulièrement pertinent à lui seul
Y : particulièrement pertinent en combinaison avec un autre document de la même catégorie
A : arrière-plan technologique
O : divulgation non-écrite
P : document intercalaire

T : théorie ou principe à la base de l'invention
E : document de brevet bénéficiant d'une date antérieure à la date de dépôt et qui n'a été publié qu'à cette date de dépôt ou qu'à une date postérieure.
D : cité dans la demande
L : cité pour d'autres raisons

& : membre de la même famille, document correspondant

1

EPO FORM 1503 12.99 (P04C14)

ANNEXE AU RAPPORT DE RECHERCHE PRÉLIMINAIRE RELATIF A LA DEMANDE DE BREVET FRANÇAIS NO. FR 0806074 FA 715697

La présente annexe indique les membres de la famille de brevets relatifs aux documents brevets cités dans le rapport de recherche préliminaire visé ci-dessus.
Les dits membres sont contenus au fichier informatique de l'Office européen des brevets à la date du 26-08-2009
Les renseignements fournis sont donnés à titre indicatif et n'engagent pas la responsabilité de l'Office européen des brevets, ni de l'Administration française

Document brevet cité au rapport de recherche	Date de publication	Membre(s) de la famille de brevet(s)	Date de publication
US 2006276698 A1	07-12-2006	AUCUN	

EPO FORM P0465

B

Décomposition en valeurs singulières

B.1 Définition

Bien que la décomposition en valeur singulière puisse être appliquée sur des matrices complexes, nous supposerons ici que toutes les matrices sont réelles.

Soit $\mathbf{M} \in \mathbb{R}^{m \times n}$ une matrice réelle de rang r tel que $r \leq \min(n,m)$. Alors il existe deux matrices orthogonales $\mathbf{U} \in \mathbb{R}^{m \times m}$ et $\mathbf{V} \in \mathbb{R}^{n \times n}$ qui vérifient :

$$\mathbf{M} = \mathbf{U} \begin{bmatrix} \Sigma_+ & 0 \\ 0 & 0 \end{bmatrix} \mathbf{V}^\top, \Sigma_+ = \begin{bmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \ddots & \\ & & & \sigma_r \end{bmatrix} \tag{B.1}$$

où $\mathbf{U}\mathbf{U}^\top = \mathbf{I}_m$, $\mathbf{V}\mathbf{V}^\top = \mathbf{I}_n$ et

$$\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r > \sigma_{r+1} = \ldots = \sigma_p = 0,\ p = \min(m,n)$$

Les valeurs $\sigma_1 \ \ldots \sigma_p$ sont alors appelées les valeurs singulières de $\mathbf{M}$ et B.1 est la décomposition en valeurs singulières (SVD).

La démonstration de l'existence de cette décomposition ainsi que des différentes propriétés ci-après pourra être trouvée notamment dans [Kat05].

B.2 Propriétés

Si on définit :

$$\Sigma = \begin{bmatrix} \Sigma_+ & 0 \\ 0 & 0 \end{bmatrix}$$

alors il est clair que B.1 peut être exprimée sous la forme

$$\mathbf{M} = \mathbf{U}\Sigma\mathbf{V}^\top = \mathbf{U}_r\Sigma_+\mathbf{V}_r^\top$$

avec $\mathbf{U}_r \in \mathbb{R}^{m \times r}$ et $\mathbf{V}_r \in \mathbb{R}^{n \times r}$. L'expression $\mathbf{M} = \mathbf{U}\Sigma_+\mathbf{V}^\top$ est appelée SVD réduite.

On choisit également la notation suivante : $\mathbf{U} = [\mathbf{U}_r \tilde{\mathbf{U}}_r] \in \mathbb{R}^{m \times m}$ et $\tilde{\mathbf{U}}_r \in \mathbb{R}^{m \times m-r}$.

Si $rang(\mathbf{M}) = r \leq min(m,n)$ alors les propriétés suivantes sont vérifiées :

1. Images et noyaux de $\mathbf{M}$ et $\mathbf{M}^\top$:

$$\begin{aligned} Im(\mathbf{M}) &= Im(\mathbf{U}_r) \quad & Ker(\mathbf{M}) &= Im(\tilde{\mathbf{V}}_r) \\ Im(\mathbf{M}^\top) &= Im(\mathbf{V}_r) \quad & Ker(\mathbf{M}^\top) &= Im(\tilde{\mathbf{U}}_r) \end{aligned}$$

2. La décomposition dyadique de $\mathbf{M}$:

$$\mathbf{M} = \sum_{i=1}^{r} \sigma_i \mathbf{u}_i \mathbf{v}_i^\top (= \mathbf{U}\Sigma\mathbf{V}^\top)$$

3. La norme de Frobenius et la norme 2 :

$$\|\mathbf{M}\|_F = \sqrt{\sigma_1^2 + \ldots + \sigma_r^2},\ \|\mathbf{M}\|_2 = \sigma_1$$

4. Équivalence des normes :

$$\|\mathbf{M}\|_2 \leq \|\mathbf{M}\|_F \leq \sqrt{p}\,\|\mathbf{M}\|_2,\ p = \min(m,n)$$

5. L'approximation par une matrice de rang inférieur. On définit la matrice $\mathbf{M}_k$ par :

$$\mathbf{M}_k = \sum_{i=1}^{k} \sigma_i \mathbf{u}_i \mathbf{v}_i^\top,\ k < r$$

alors avec $k = rang(\mathbf{M}_k)$ et

$$\min_{rang(\mathbf{B})=k} \|\mathbf{M} - \mathbf{B}\|_2 = \|\mathbf{M} - \mathbf{M}_k\|_2 = \sigma_{k+1}$$

C

Eléments de calcul tensoriel

Cette annexe ne constitue qu'une très brève introduction au calcul tensoriel, afin de présenter quelques outils mathématiques utiles. Une description plus complète pourra notamment être trouvée dans [Gro00]

C.1 Définition

Soit V un espace vectoriel de dimension n sur un corps K. L'espace dual V^* est l'espace vectoriel formé de toutes les formes linéaires

$$f : V \to K$$

L'espace V^* est aussi de dimension n. Les éléments de V et V^* sont appelés respectivement vecteurs et covecteurs. Un tenseur est une application multilinéaire :

$$T : \underbrace{V^* \times \ldots \times V^*}_{h} \times \underbrace{V \times \ldots \times V}_{k} \to K$$

Un tenseur T associe alors à k vecteurs $v_1, \ldots, v_k$ et h covecteurs $w_1, \ldots, w_h$ à un scalaire $T(w_1, \ldots, w_h, v_1, \ldots, v_k)$. La valence du tenseur est le couple (h,k). L'ordre du tenseur correspond à la somme $h+k$.

C.2 Tenseurs euclidiens

Dans ce document, tous les tenseurs seront définis à partir de l'ensemble des réels $\mathbb{R}$. La structure euclidienne de cet espace permet des simplifications lors de la manipulation des tenseurs.

En effet, dans un espace euclidien, l'existence d'un produit scalaire réel g fournit des propriétés particulières aux tenseurs. Il permet notamment d'établir un isomorphisme canonique entre l'espace V et l'espace V^* associant une unique forme linéaire f à tout vecteur v :

$$f : V \to K \text{ avec } \forall u \in V \;\; f(u) = g(v,u)$$

Via cet isomorphisme, on peut alors assimiler tout élément de V^* à un élément de V. D'une manière générale, cela permet de ne plus distinguer les vecteurs et les covecteurs dans la définition d'un tenseur. Dans ces conditions,

un tenseur de valence (h,k) peut aussi bien être vu comme un tenseur de valence $(h+k,0)$ ou $(0,h+k)$. L'ordre $(h+k)$ devient une caractéristique suffisante pour catégoriser tout tenseur construit sur V.

En considérant une base particulière de V, on peut alors représenter le tenseur T par une grandeur indicée $h+k$ fois où chacun des indices va de 1 à n : $(T_{i,j,k,\ldots})_{1\leq i,j,k\ldots\leq n}$.

On appelle composante chacun des nombres $T_{i,j,k,\ldots}$

Par la suite, tous les tenseurs seront supposés euclidiens.

C.3 Opérations sur les tenseurs

C.3.1 Produit tensoriel

Soit A un tenseur d'ordre h et B un tenseur d'ordre k, le produit tensoriel de A par B, noté $A\otimes B$, est égal au tenseur C d'ordre $h+k$ dont les composantes sont définies par :

$$C_{i_1,\ldots,i_h,j_1,\ldots,j_k} = A_{i_1,\ldots,i_h}B_{j_1,\ldots,j_k}$$

C.3.2 Contraction

Soit A un tenseur d'ordre h et i et j deux indices tels que $1\leq i\leq j\leq h$. La contraction du tenseur A sur les indices i et j est égale au tenseur B d'ordre $h-2$ dont les composantes sont définies par :

$$B_{1,\ldots,i-1,i+1,\ldots,j-1,j+1,\ldots,h} = \sum_{k=1}^{n} A_{1,\ldots,i-1,k,i+1,\ldots,j-1,k,j+1,\ldots,h}$$

C.3.3 Produit tensoriel contracté

Soit A un tenseur d'ordre h et B un tenseur d'ordre k, le produit tensoriel contracté 1 fois de A par B, noté $A\overline{\otimes}B$, est égal au tenseur C d'ordre $h+k-2$, calculé à partir du produit tensoriel de A par B ayant subi une contraction, par convention effectué sur les indices correspondant au dernier indice de A et au premier indice de B.

Le produit tensoriel contracté k fois de A par B, noté $A\overline{\overline{\otimes}}B$, est défini de manière analogue.

C.3.4 Remarque

L'espace des tenseurs d'ordre n définis sur V est un espace euclidien dont le produit scalaire correspond au produit tensoriel contracté n fois. Plus généralement, le produit tensoriel contracté peut être utilisé comme produit de dualité pour les tenseurs.

C.4 Gradient

C.4.1 Dérivée directionnelle ou dérivée au sens de Gâteaux

Soient E un espace vectoriel normé, f une fonction définie sur E à valeurs dans un espace vectoriel normé F, x et h deux éléments de E. La dérivée de f en x dans la direction h est, si elle existe, la dérivée en 0 de la fonction de la variable réelle $t \to f(u+th)$:

$$D_h f(x) = \lim_{t \to 0} \frac{f(x+th) - f(x)}{t}$$

C.4.2 Gradient

Si la fonction f est différentiable en x, alors elle admet des dérivées en ce point dans toutes les directions. Il existe donc une fonction linéaire $g : E \to F$ telle que $\forall h, g(h) = D_h f(x)$. Si de plus E est un espace vectoriel euclidien, il existe donc un élément $\nabla_x f \in E \times F$ tel que :

$$D_h f(x) = \langle \nabla_x f | h \rangle$$

où $\langle .|. \rangle$ représente un produit de dualité.

Si l'on choisit une base de E, l'expression de ∇_x f peut alors se mettre sous la forme :

$$\nabla_x f = \frac{\partial f}{\partial x} = \left[\frac{\partial f}{\partial x_1} \cdots \frac{f}{\partial x_n} \right]^\top$$

C.4.3 Gradient d'une fonction composée

Soit $g \circ f$ une fonction composée de $E \to G$ avec $f : E \to F$ et $g : F \to G$, f et g étant supposées dérivables. On suppose que E, F et G sont des espaces vectoriels euclidiens. La dérivée de $g \circ f$ en x dans la direction h est égale à :

$$\begin{aligned} D_h g \circ f(x) &= \lim_{t \to 0} \frac{g(f(x+th)) - g(f(x))}{t} \\ &= \lim_{t \to 0} \frac{g(f(x) + tD_h f(x)) - g(f(x))}{t} \\ &= D_{D_h f(x)} g(f(x)) \end{aligned}$$

On peut également écrire :

$$\begin{aligned} D_{D_h f(x)} g(f(x)) &= \langle \nabla_{f(x)} g | D_h f(x) \rangle \\ &= \langle \nabla_{f(x)} g | \langle \nabla_x f | h \rangle \rangle \end{aligned}$$

On obtient alors :

$$\nabla_x g \circ f = \langle \nabla_{f(x)} g | \nabla_x f \rangle$$

D

Algorithme Expectation-Maximisation

Soit X l'ensemble des variables observées, et Z l'ensemble des variables latentes. X et Z se décomposent le plus souvent en ensemble d'éléments indépendants et identiquement distribués (i.i.d.), et X peut être écrit sous la forme $X = \{X_1, X_2, \ldots, X_n\}$, où les X_i sont les variables (i.i.d.) et les données observées, $x = \{x_1, x_2, \ldots, x_n\}$, sont les valeurs prises par X. Le but est d'estimer la probabilité qu'un modèle ait généré ces données : $P(x, z|\mathbf{w})$ où $\mathbf{w}$ représente le vecteur des paramètres du dit modèle.

Si Z pouvait être observé, le problème d'estimation du maximum de vraisemblance [JB02] reviendrait à maximiser la log-vraisemblance complète :

$$\log P(x, z|\mathbf{w})$$

L'utilisation du logarithme permet notamment de décomposer la probabilité $P(x, z|\mathbf{w})$ lorsque celle-ci peut s'écrire sous la forme d'un produit de facteurs. Étant donné que Z n'est pas observé, la probabilité de la donnée x est une probabilité marginale, et la log-vraisemblance incomplète s'écrit :

$$\log P(x|\mathbf{w}) = \log \sum_z P(x, z|\mathbf{w})$$

où la marginalisation est effectuée par une somme dans le cas discret (ou une intégration dans le cas où z est continu). Toutefois, à ce stade, le problème d'estimation n'est pas découplé.

Étant donné que Z n'est pas observé, la log-vraisemblance complète est une quantité aléatoire et ne peut pas être maximisée directement. Mais il est possible de supprimer cet aléatoire en moyennant sur z en utilisant la distribution $q(z|x)$. On peut alors définir la log-vraisemblance complète :

$$l(\mathbf{w}, x, z)_q = \sum_z q(z|x, \mathbf{w}) \log P(x, z|\mathbf{w})$$

qui est une fonction déterministe de $\mathbf{w}$. La log-vraisemblance complète attendue est une fonction linéaire de la log-vraisemblance complète. De plus, si q est bien choisie, il est alors possible que la log-vraisemblance complète attendue soit un substitut intéressant de la vraie log-vraisemblance. Même s'il est difficile d'imaginer que maximiser ce substitut va permettre de trouver les paramètres $\mathbf{w}$ qui maximisent la vraisemblance, il est possible d'espérer une amélioration vis-à-vis de valeurs initiales de $\mathbf{w}$ et ensuite d'itérer le processus. Il s'agit de l'idée maîtresse de l'algorithme EM.

Nous allons montrer que la distribution $q(z|x)$ peut être utilisée pour calculer une borne inférieure de la log-

vraisemblance. En effet, on a :

$$\begin{aligned} l(\mathbf{w},x) &= \log P(x|\mathbf{w}) \\ &= \log \sum_z P(x,z,|\mathbf{w}) \\ &= \log \sum_z q(z|x) \frac{P(x,z,|\mathbf{w})}{q(z|x)} \\ &= \sum_z q(z|x) \log \frac{P(x,z,|\mathbf{w})}{q(z|x)} \\ &= L(q,\mathbf{w},x) \end{aligned}$$

où la dernière ligne définit $L(q,\mathbf{w},x)$, une fonction auxiliaire. Dans l'équation ..., l'inégalité de Jensen a été utilisée, étant donné la concavité de la fonction logarithme. Ainsi pour une distribution $q(z,x)$ arbitraire, la fonction auxiliaire $L(q,\mathbf{w},x)$ est une borne inférieure de la log-vraisemblance. L'algorithme EM est un algorithme d'ascension appliqué sur la fonction $L(q,\mathbf{w},x)$. A l'itération $t+1$, $L(q,\mathbf{w}_t,x)$ est d'abord maximiser selon q pour obtenir q_{t+1}, et ensuite $L(q_{t+1},\mathbf{w},x)$ est maximiser selon $\mathbf{w}$, ce qui permet de calculer $\mathbf{w}_{t+1}$. Il s'agit des 2 étapes classiques de l'algorithme :

$$\begin{aligned} \text{Etape E} : q_{t+1} &= \text{argmax}_q \, L(q,\mathbf{w}_t,x) \\ \text{Etape M} : \mathbf{w}_{t+1} &= \text{argmax}_{\mathbf{w}} \, L(q_{t+1},\mathbf{w},x) \end{aligned}$$

Pour comprendre le nom de chaque étape, on peut remarquer que l'étape M peut être vue comme une maximisation de la log-vraisemblance complète attendue. En effet, la borne inférieure $L(q,\mathbf{w},x)$ peut se décomposer en :

$$\begin{aligned} L(q,\mathbf{w},x) &= \sum_z q(z|x) \log \left[P(x,z|\mathbf{w})/q(z|x)\right] \\ &= \sum_z q(z|x) \log P(x,z|\mathbf{w}) - \sum_z q(z|x) \log q(z|x) \\ &= l(\mathbf{w},x,z)_q - \sum_z q(z|x) \log q(z|x) \end{aligned}$$

Le second terme étant indépendant de $\mathbf{w}$, maximiser $L(q,\mathbf{w},x)$ selon $\mathbf{w}$ est équivalent à maximiser $l(\mathbf{w},x,z)_q$ selon $\mathbf{w}$.

Pour l'étape E, le problème de maximisation se résout simplement car la solution se résume au choix $q_{t+1}(z|x) = P(z|x,\mathbf{w}_t)$. Ce choix correspond en effet au maximum de $L(q,\mathbf{w},x)$ car on a :

$$\begin{aligned} L(P(z|x,\mathbf{w}_t),\mathbf{w}_t,x) &= \sum_z P(z|x,\mathbf{w}_t) \left[\log P(x,z|\mathbf{w}_t)/P(z|x,\mathbf{w}_t)\right] \\ &= \sum_z P(z|x,\mathbf{w}_t) \log P(x|\mathbf{w}_t) \\ &= \log P(x|\mathbf{w}_t) \\ &= l(\mathbf{w}_t,x) \end{aligned}$$

Étant donné que $l(\mathbf{w}_t,x)$ est une borne supérieure de $L(P(z|x,\mathbf{w}_t),\mathbf{w}_t,x)$, le choix $q(z|x) = P(z|x,\mathbf{w}_t)$ maximise bien la fonction $L(q(z|x),\mathbf{w}_t,x)$. Ainsi, l'algorithme EM maximise la log-vraisemblance complète attendue en fonction des paramètres du modèle, puis ce nouveau modèle amélioré est utilisé pour obtenir une meilleure distribution pour calculer à nouveau la log-vraisemblance complète attendue pour la prochaine itération. Cette procédure est répétée jusqu'à convergence de $L(q,\mathbf{w},x)$. La question restante est que l'étape M ne maximise qu'une borne inférieure de la log vraisemblance. Cependant, l'étape E garantit que $l(\mathbf{w}_t,x) = L(q_{t+1},\mathbf{w}_t,x)$ et donc que l'étape M va effectivement maximiser $l(\mathbf{w}_t,x)$. L'algorithme EM est donc bien un algorithme de maximalisation de la log-vraisemblance.

Bibliographie

[ACDR06] J. Altet, W. Claeys, S. Dilhaire, and A. Rubio. Dynamic surface temperature measurements in ics,. Proceedings of the IEEE, 94 :1519 – 1533, August 2006.

[ADJC+04] R. Aubry, C. Dua, J.-C.Jacquet, F. Lemaire, P. Galtier, B. Dessertenne, Y. Cordier, M.-A. Diforte-Poisson, and S. L. Delage. Temperature measurement by micro-raman scattering spectroscopy in the active zone of algan/gan high-electron-mobility transistors. European Physical Journal Applied Physics, 27 :293–296, July 2004.

[AHU58] K. Arrow, L. Hurwicz, and H. Uzawa. Studies in Nonlinear Programming. Stanford Univ. Press, 1958.

[Aka73] H. Akaike. Information theory as an extension of the maximum likelihood principle. In B.N. Petrov and F. Csaki, editors, Second International Symposium on Information Theory, pages 267–281, Akademiai Kiado, Budapest, 1973.

[ASG01] A. C. Antoulas, D. C. Sorensen, and S. Gugercin. A survey of model reduction methods for large-scale systems. Contemporary Mathematics, 280 :193–219, 2001.

[BB08] L. Bottou and O. Bousquet. The tradeoffs of large scale learning. In J.C. Platt, D. Koller, Y. Singer, and S. Roweis, editors, Advances in Neural Information Processing Systems 20, pages 161–168. MIT Press, Cambridge, MA, 2008.

[BLO02] J.V. Burke, A.S. Lewis, and M.L. Overton. Two numerical methods for optimizing matrix stability. SIAM J. Optimization 15, 2002.

[BLO05] J.V. Burke, A.S. Lewis, and M.L. Overton. A robust gradient sampling algorithm for nonsmooth, nonconvex optimization. SIAM J. Optimization, 15 :751–779, 2005.

[Bon06] F. Bonnans. Optimisation Continue. Dunod, Paris, 2006.

[Bot04] L. Bottou. Stochastic learning. In Olivier Bousquet and Ulrike von Luxburg, editors, Advanced Lectures on Machine Learning, Lecture Notes in Artificial Intelligence, LNAI 3176, pages 146–168. Springer Verlag, Berlin, 2004.

[BSF94] Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. IEEE Transactions on Neural Network, 5 :157–166, 1994.

[BV04] S. Boyd and L. Vandenberghe. Convex Optimization. Cambridge University Press, March 2004.

[CGGR05] S. Canu, Y. Grandvalet, V. Guigue, and A. Rakotomamonjy. Svm and kernel methods matlab toolbox. Perception Systèmes et Information, INSA de Rouen, Rouen, France, 2005.

[Che99] B. Cheron. Transfert Thermique. Ellipses, 1999.

[CL01] C.-C. Chang and C.-J. Lin. LIBSVM : a library for support vector machines, 2001.

[CM05] N. L. C. Chui and J.M. Maciejowski. Subspace identification – a markov parameter approach. International Journal of Control, 78(17) :1412–1436, 2005.

[Dec96] R. Dechter. Bucket elimination : A unifying framework for probabilistic inference. Proc. Twelthth Conf. on Uncertainty in Artificial Intelligence, pages 211–219, 1996.

[DMS+02] G. Dreyfus, J.-M. Martinez, M. Samuelides, M. B. Gordon, F. Badran, S. Thiria, and L. Hérault. Réseaux de neurones, méthodologie et applications. Eyrolles, 2002.

[DMS+08] G. Dreyfus, J.-M. Martinez, M. Samuelides, F. Badran M.B. Gordon, and S. Thiria. Apprentissage statistique. Eyrolles, 2008.

[EDG+05] Y. Ezzahri, S. Dilhaire, S. Grauby, J.M. Rampnoux, W. Claeys, Y. Zhang, G. Zeng, and A. Shakouri. Study of thermomechanical properties of si/sige superlattices using femtosecond transient thermoreflectance technique. Applied Physics Letters, (87), 2005.

[ESM05] M. Espinoza, J. A. K. Suykens, and B. De Moor. Kernel based partially linear models and nonlinear identification. IEEE Transactions on Automatic Control, 50(10) :1602–1606, 2005.

[Flo] Flomerics. Flotherm. www.flomerics.fr/flotherm/.

[Fou22] J. Fourier. Théorie analytique de la chaleur. Firmin Didot, père et fils, 1822.

[FPSM09] T. Falck, K. Pelckmans, J.A.K. Suykens, and B. De Moor. Identication of wiener-hammerstein systems using ls-svms. In Proceedings of 15th IFAC Symposium on System Identification (SYSID), pages 820–825, 2009.

[FTH+03] C. Filloy, G. Tessier, S. Holé, G. Jerosolimski, and D. Fournie. The contribution of thermoreflectance to high resolution thermal mapping. Sensor Review, 23(1) :35–39, 2003.

[GBD92] S. Geman, E. Bienenstock, and R. Doursat. Neural networks and the bias/variance dilemma. Neural Computation, 4 :1–58, 1992.

[GDH+01] A. Gretton, A. Doucet, R. Herbrich, P.J.W. Rayner, and B. Scholkopf. Support vector regression for black-box system identification. In Proceedings of the 11th IEEE Signal Processing Workshop on, pages 341 –344, 2001.

[GJP95] F. Girosi, M. Jones, and T. Poggio. Regularization theory and neural networks architectures. Neural Computation, 7 :219–269, 1995.

[Gon05] E. Goncalvès. Résolution numérique, discrétisation des EDP et EDO. Cours de l'Institut National Polytechnique de Grenoble, 2005.

[GPSM05] I. Goethals, K. Pelckmans, J. A. K. Suykens, and B. De Moor. Identification of mimo hammerstein models using least squares support vector machines. Automatica, 41(7) :1263–1272, 2005.

[Gro00] P. Gros. Introduction à l'algèbre tensorielle. Number 238. 2000.

[Gus00] F. Gustafson. Adaptive filtering and change detection. John Wiley & Sons, Ltd, 2000.

[Heb49] D. O. Hebb. The Organization of Behavior : A Neuropsychological Theory. Wiley, New York, June 1949.

[HSW89] K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. Neural Network, 2(5) :359–366, 1989.

[HSWA] K. Hornik, M. Stinchcombe, H. White, and P. Auer. Degree of approximation results for feed-forward networks approximating unknown mappings and their derivatives. Neural Computation.

[HTF09] T. Hastie, R. Tibshirani, and J. Friedman. The Elements of Statistical Learning : Data Mining, Inference and Prediction. Springer Verlag, New York, 2009.

[HUL93] J.-B. Hiriart-Urruty and C. Lemarechal. Convex Analysis and Minimization Algorithms I : Fundamentals (Grundlehren Der Mathematischen Wissenschaften). Springer, 1993.

[HY03] Mark H. Hansen and Bin Yu. Minimum description length model selection criteria for generalized linear models. Lecture Notes-Monograph Series, 40 :145–163, 2003.

[Iva76] V. V. Ivanov. The theory of approximate methods and their application to the numerical solution of singular integral equations. Nordhoff International, 1976.

[JB02] M.I. Jordan and C. Bishop. Introduction to graphical model, 2002.

[JLO90] F. V. Jensen, S.L. Lauritzen, and K.G. Olesen. Bayesian updating in recursive graphical models by local computation. Computational Statistics Quarterly, pages 269–282, 1990.

[Kat05] T. Katayama. Subspace methods for system identification. Springer, 2005.

[Kay93] S. M. Kay. Fundamentals of Statistical Signal Processing, Volume I : Estimation Theory. Prentice Hall PTR, March 1993.

[KYM07] Y. Kawahara, T. Yairi, and K. Machida. A kernel subspace method by stochastic realization for learning nonlinear dynamical systems. In B. Schölkopf, J. Platt, and T. Hoffman, editors, Advances in Neural Information Processing Systems 19, pages 665–672. MIT Press, Cambridge, MA, 2007.

[Lar90] W. E. Larimore. Canonical variate analysis in identification, filtering and adaptive control. In In Proceedings of 29th IEEE Conference on Decision and Control, pages 596–604, 1990.

[Lau08] F. Lauer. Machines à Vecteurs de Support et Identification de Systèmes Hybrides. PhD thesis, Université Henri Poincarré - Nancy I, 2008.

[LB02] S. L. Lacy and D. S. Bernstein. Subspace identification with guaranteed stability using constrained optimization. Proc. American Control Conference, 2002.

[LB03] S. L. Lacy and D. S. Bernstein. Subspace identification with guaranteed stability using constrained optimization. IEEE Transactions on Automatic Control, 2003.

[LeC85] Y. LeCun. Une procédure d'apprentissage pour réseau à seuil asymétrique. Proceedings of Cognitiva 85, pages 599–604, 1985.

[Lju00] L. Ljung. System Identification Toolbox - for use with Matlab. Mathworks, Inc., 2000.

[Lju02] L. Ljung. System Identification : Theory for the User. PTR Prentice Hall, second edition, 2002.

[LL95] P. Lindskog and L. Ljung. Tools for semiphysical modelling. International Journal of Adaptive Control and Signal Processing, 9(6) :509–523, November 1995.

[Mac95] J. M. Maciejowski. Guaranteed stability with subspace methods. Systems and Control Letters, 26(2) :153–156, 1995.

[Mal] C. L. Mallows. Some comments on C_p. Technometrics.

[Mer09] J. Mercer. Functions of positive and negative type and their connection with the theory of integral equations. Philos. Trans. Roy. Soc. London, 1909.

[MGC08] G. Mallet, G. Gasso, and S. Canu. New methods for the identification of a stable subspace model for dynamical systems. In Proceedings of 8th IEEE Workshop Machine Learning for Signal Processing, 2008.

[Mie99] K. Miettinen. Nonlinear Multiobjective Optimization, volume 12 of International Series in Operations Research and Management Science. Kluwer Academic Publishers, Dordrecht, 1999.

[MLH+06] G. Mallet, P. Leray, H.Polaert, C. Tolant, and P. Eudeline. Dynamic compact thermal model with neural networks for radar applications. In Therminic's, 2006.

[MLP07] G. Mallet, P. Leray, and H. Polaert. Méthodes statistiques et modèles thermiques compacts. In Actes de la Conférence Extraction et Gestion de la Connaissance EGC, 2007.

[Mor84] V. A. Morozov. Methods for Solving Incorrectly Posed Problems. Springer-Verlag, New York, 1984.

[MP43] W. S. Mcculloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. Bulletin of Mathematical Biophysic, 5 :115–133, 1943.

[MP69] M. L. Minsky and S. A. Papert. Perceptrons. The MIT Press, December 1969.

[MP09] G. Mallet and H. Polaert. Repositionnement et recalage d'images infrarouges. 2009.

[MRRÁCV+06] M. Martínez-Ramón, J. L. Rojo-Álvarez, G. Camps-Valls, J. Muñoz-Marí, A. Navia-Vázquez, E. Soria-Olivas, and A. R. Figueiras-Vidal. Support vector machines for nonlinear kernel arma system identification. IEEE Transactions on Neural Networks, 17(6) :1617–1622, 2006.

[MS09] A. Marconato and J. Schoukens. Identification of wiener-hammerstein benchmark data by means of support vector machines. In In Proceedings of 15th IFAC Symposium on System Identification (SYSID), pages 816–819, 2009.

[Mur02] K. P. Murphy. Dynamic Bayesian Networks : Representation, Inference and Learning. PhD thesis, Cambridge University, University of Pennsyvania, 2002.

[Muz06] Y. S. Muzychka. Influence coefficient method for calculating discrete heat source temperature on finite convectively cooled substrates. In IEEE Transactions on Components and Packaging Technologies, volume 9, pages 636–643, 2006.

[Nav22] C. L. M. H. Navier. Mémoire sur les lois du mouvement des fluides. Académie des Sciences, 1822.

[NRPH00] M. Norgaard, O. Ravn, N. K. Poulsen, and L. K. Hansen. Neural Networks for Modelling and Control of Dynamic Systems. Springer-Verlag, 2000.

[NW06] J. Nocedal and S. J. Wright. Numerical Optimization. Springer, second edition, 2006.

[NWL+04] P. Naïm, P-H. Wuillemin, P. Leray, O. Pourret, and A. Becker. Réseaux bayésiens. Eyrolles, Paris, 2004.

[OM94] P. Van Overschee and B. De Moor. N4sid : subspace algorithms for the identification of combined deterministic-stochastic systems. Automatica, 30 :75–93, 1994.

[OM95] P. Van Overschee and B. De Moor. A unifying theorem for three subspace system identification algorithms. Automatica, 31(12) :1853–1864, 1995.

[OM96] . Van Overschee and B. De Moor. Subspace identification for linear systems : Theory, Implementation and Applications. Kluwer Academic Publishers, 1996.

[Pea88] J. Pearl. Probabilistic reasoning in intelligent systems : networks of plausible inference. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1988.

[Pek04] K. M. Pekpe. Identification par les techniques des sous-espaces - application au diagnostic. PhD thesis, Institut National Polytechnique de Lorraine, 2004.

[PM14] M. Planck and M. Masius. The theory of heat radiation. P. Blakiston's Son and Co, 1914.

[PP08] K. B. Petersen and M. S. Pedersen. The matrix cookbook, oct 2008. Version 20081110.

[Pre98] L. Prechelt. Early stopping - but when. In Neural Networks : Tricks of the Trade, volume 1524 of LNCS, chapter 2, pages 55–69. Springer-Verlag, 1998.

[RdB04] L. Ralaivola and F. d'Alche Buc. Dynamical modeling with kernels for nonlinear time series prediction. In Sebastian Thrun, Lawrence Saul, and Bernhard Schölkopf, editors, Advances in Neural Information Processing Systems 16. MIT Press, Cambridge, MA, 2004.

[rdpA] Harvard Thermal (racheté depuis par Ansys). Tas, thermal analysis system.

[RH85] C.R. Johnson R.A. Horn. Matrix Analysis. Cambridge University Press, 1985.

[RHW86] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning internal representations by error propagation. Parallel Distributed Processing, 1 :318–362, 1986.

[Ris78] J. Rissanen. Modelling by the shortest data description. Automatica, 14 :661–675, 1978.

[Ros58] F. Rosenblatt. The perceptron : a probabilistic model for information storage and organization in the brain. Psychological Review, 65(6) :386–408, November 1958.

[SBG08] S. Siddiqi, B. Boots, and G. Gordon. A constraint generation approach to learning stable linear dynamical systems. In J.C. Platt, D. Koller, Y. Singer, and S. Roweis, editors, Advances in Neural Information Processing Systems 20, pages 1329–1336. MIT Press, Cambridge, MA, 2008.

[Sch78] G. Schwarz. Estimating the dimension of a model. Annals of Statistics, 6 :461–464, 1978.

[SGB+02] J.A.K. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor, and J. Vandewalle. Least squares support vector machines. World Scientific, 2002.

[SLV00] J.A.K. Suykens, L. Lukas, and J. Vandewalle. Sparse approximation using least squares support vector machines. In IEEE International Symposium on Circuits and Systems ISCAS'2000, 2000, pages 757–760, 2000.

[SS01] B. Scholkopf and A. J. Smola. Learning with Kernels : Support Vector Machines, Regularization, Optimization, and Beyond. MIT Press, Cambridge, MA, USA, 2001.

[Stu98] J. F. Sturm. Using sedumi 1.02, a matlab toolbox for optimization over symmetric cones, 1998.

[Suy00] J. A. K. Suykens. Recurrent least squares support vector machines. IEEE Transactions on Circuits and Systems-I, 47 :1109–1114, 2000.

[SV99] J. A. K. Suykens and J. Vandewalle. Least squares support vector machine classifiers. Neural Processing Letters, 9(3) :293–300, June 1999.

[SVM01] J.A.K. Suykens, J. Vandewalle, and B. De Moor. Optimal control by least squares support vector machines. Neural Networks, 14 :23–35, 2001.

[TA77] A.N. Tikhonov and V.Y. Arsenin. Solutions of Ill-posed Problems. W.H. Winston ed., 1977.

[TC02] C. Tolant and J. Cordier. Method and device to measure the temperature of microwave components, 2002.

[Vap95] V. Vapnik. The Nature of Statistical Learning Theory. New York : Springer Verlag, 1995.

[Vap98] V. Vapnik. Statistical Learning Theory. Willey, 1998.

[Ver94] M. Verhaegen. Identification of the deterministic part of mimo state space models given in innovations form from input-output data. Automatica, 30(1) :61–74, 1994.

[Vib02] M. Viberg. Subspace-based state-space system identification. Circuits, Systems and Signal Processing, Volume 21(1) :23–37, 2002.

[Wah90] G. Wahba. Spline models for observational data. Series in Applied Mathematics, 59, 1990.

[Wer74] P. J. Werbos. Beyond Regression : New Tools for Prediction and Analysis in the Behavioral Sciences. Harvard University, 1974.

[Wer90] P. J. Werbos. Backpropagation through time : what it does and how to do it. Proceedings of the IEEE, 78(10) :1550–1560, 1990.

[WG04] E. Wernholt and S. Gunnarsson. Nonlinear grey-box identification of industrial robots containing flexibilities. Technical Report LiTH-ISY-R-2641, Department of Electrical Engineering, Linköping University, SE-581 83 Linköping, Sweden, November 2004.

[WH60] B. Widrow and M. E. Hoff. Adaptive switching circuits. IRE WESCON Convention Record, 4 :96–104, 1960.

[WZ89] R. J. Williams and D. Zipser. A learning algorithm for continually running fully recurrent neural networks. Neural Computation, 1 :270–280, 1989.

[YFL05] Z. Yu, X. Fu, and Y. Li. Online support vector regression for system identification. In Advances in Natural Computation, volume 3611, pages 627–630, 2005.

Résumé

Cette thèse s'intéresse à l'application des méthodes d'apprentissage statistique pour la prédiction de température d'un composant électronique présent dans un radar. On étudie un cas simplifié des systèmes réels, le système étudié se limitant à un seul composant monté sur un système de refroidissement réduit. Le premier chapitre est consacré à la modélisation thermique. Après avoir présenté les principaux modes de transmission de l'agitation thermique, les modèles analytiques et numériques qui en découlent sont étudiés. En utilisant cette connaissance, le deuxième chapitre propose de choisir dans les méthodes de mesures les plus adaptées aux spécifications et aux contraintes de l'application choisie. Une fois que les bases de données ont été établies, nous pouvons utiliser dans le troisième chapitre les techniques de l'apprentissage statistique pour construire un modèle dynamique. Après un bref rappel sur les tenants et les aboutissants de la modélisation statistique, quatre familles de méthodes seront présentées : les modèles linéaires, les réseaux de neurones, les réseaux bayésiens dynamiques et les machines à vecteur support (SVM). Enfin, le quatrième chapitre est l'occasion de présenter une méthode de modélisation originale. En effet, après avoir détaillé la mise en oeuvre des méthodes d'identification de représentation d'état, nous verrons comment prendre en compte des a priori théoriques au cours de l'apprentissage de ce type de modèle, à savoir une contrainte de stabilité.

Abstract

This thesis is focused on the application of statistical learning methods for the temperature prediction of an electronic component embedded in a radar. We study a simplified case of real systems, the system under study is limited to a single component mounted on a reduced cooling system. The first chapter is devoted to heat transfer modelisation. After presenting the major mechanisms of thermal agitation transmission, analytical and numerical models are studied. Using this knowledge, the second chapter offers a survey on the methods of temperature measurement, choosing the fittest according to the specifications and the constraints of the chosen application. Once databases have been established, we can use in the third chapter statistical learning techniques to build a dynamic model. After a brief reminder about the ins and outs of statistical modeling, four families of methods will be presented : linear models, neural networks, dynamic bayesian networks and support vector machines (SVM). The fourth chapter is an opportunity to present a novel method of modeling. Indeed, after a presentation of the methods for the identification of state representation, we see how to take into account theoretical apriorism during learning of this model type, ie a stability constraint.