



**HAL**  
open science

# Capteurs intelligents : quelles méthodologies pour la fusion de données embarquées ?

Aurelien Valade

► **To cite this version:**

Aurelien Valade. Capteurs intelligents : quelles méthodologies pour la fusion de données embarquées?. Micro et nanotechnologies/Microélectronique. INSA de Toulouse, 2017. Français. NNT : 2017ISAT0007 . tel-01592864

**HAL Id: tel-01592864**

**<https://theses.hal.science/tel-01592864>**

Submitted on 25 Sep 2017

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



# THÈSE

En vue de l'obtention du

## DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par :

*l'Institut National des Sciences Appliquées de Toulouse (INSA de Toulouse)*

---

---

Présentée et soutenue le 18/05/2017 par :

**AURÉLIEN VALADE**

**Capteurs intelligents : quelles méthodologies pour la fusion  
de données embarquée?**

---

---

**École doctorale et spécialité :**

*GEET : Micro et Nanosystèmes*

**Unité de Recherche :**

*Laboratoire d'Analyse et d'Architecture des Systèmes (LAAS-CNRS)*

**Président du Jury :**

THIERRY CAMPS

Professeur des Universités

Université Paul Sabatier

**Rapporteurs :**

VIRGINIE BLONDEAU-PATISSIER

Maître de Conférences (HdR)

Institut FEMTO-ST

CAMEL TANOUGAST

Professeur des Universités

Université de Lorraine

**Examineurs :**

RÉMI FLOQUET

Ingénieur R&D

RF Innovation

**Membres invités :**

JEAN MILPIED

Manager R&D

TE Connectivity

PASCAL ACCO

Maître de Conférences

INSA Toulouse

**Directeurs de Thèse :**

JEAN-YVES FOURNIOLS

Professeur des Universités

LAAS-CNRS

GEORGES SOTO-ROMERO

Maître de Conférences

LAAS-CNRS



# Résumé

Les travaux de thèse présentés ici sont le fruit d'un travail collaboratif effectué entre le *LAAS-CNRS* de Toulouse et l'entreprise *MEAS-France / TE Connectivity* sur une durée de trois ans. L'objectif est ici de mettre en place une méthodologie permettant la réalisation de capteurs embarqués intelligents capables d'estimer des paramètres physiques à partir de la fusion de données multi-physiques, cette stratégie devant permettre l'intégration de capteurs, jusqu'alors réservés aux mesures en laboratoire dans des environnement défavorables, au sein de systèmes embarqués disposant de faibles puissances de calculs.

Après une exploration des méthodes liées à la modélisation, l'estimation de paramètres et les filtres de Kalman, nous détaillons différentes solutions existantes sur lesquelles nous appuyer pour répondre à la problématique de la fusion de données multi-physiques, pour les systèmes linéaires, avec le filtre de Kalman, les systèmes non-linéaires avec les filtres de Extended Kalman Filter, et Unscented Kalman Filter. Puis nous synthétisons un filtre pour les systèmes hybrides, présentant par exemple une évolution linéaire et une mesure non-linéaire, utilisant le meilleur des deux parties afin de tendre vers le meilleur compromis complexité/précision.

Une fois la méthode d'estimation sélectionnée, nous nous attardons sur les problématiques liées à la puissance de calcul et à la complexité algorithmique, pour chercher les optimisations pouvant être mises en place afin de garantir le fonctionnement sur un système embarqué disposant d'une faible puissance de calcul.

Nous exposons ensuite l'application de cette méthodologie sur le cas d'étude du capteur UQS, commercialisé par *TE Connectivity*, utilisant la spectroscopie proche infrarouge pour déterminer la concentration d'urée en solution dans le cadre de la régulation des systèmes de dépollution des oxydes d'azote sur les moteurs diesel. Après une présentation des principes de fonctionnement, nous présentons le modèle développé que nous avons dressé afin de représenter le système, de simuler son comportement et d'effectuer la fusion de mesures permettant d'extraire la concentration désirée. Au cours de cette étape, nous abordons les problématiques liées à la calibration du modèle, ainsi qu'à la compensation de dérives, qu'elles soient liées aux conditions d'utilisation ou au vieillissement des composants.

En s'appuyant sur cette réflexion, nous dressons finalement les modèles hybrides nous permettant d'aborder les cas de fonctionnement nominaux, ainsi que les recalibrations du modèle au cours de la vie du produit. Après quoi, nous présentons les résultats obtenus, d'abord sur des données simulées, puis sur des données capturées sur un système réel, avant de proposer des pistes d'amélioration de la méthodologie en se basant sur des modèles tabulés en « boîte noire » qui permettent de faciliter l'étape de calibration tout en diminuant la charge de calcul.

Finalement, nous généralisons la méthodologie par le biais d'une problématique de capture de mouvement, pour conclure sur les possibilités et les limites de la solution que nous avons proposée.



# Abstract

The work detailed in this document is the result of a collaborative effort of the LAAS-CNRS in Toulouse and MEAS-France / TE Connectivity during a period of three years. The goal here is to develop a methodology to design smart embedded sensors with the ability to estimate physical parameters based on multi-physical data fusion. This strategy tends to integrate sensors technologies, currently dedicated to lab measurements, in low powered embedded systems working in imperfect environments.

After exploring model oriented methods, parameters estimations and Kalman filters, we detail various existing solutions upon which we can build a valid response to multi-physical data fusion problematics, for linear systems with the Kalman Filter, and for non-linear systems with the Extended Kalman Filter and the Unscented Kalman Filter. Then, we will synthesize a filter for hybrid systems, having a linear evolution model and a non-linear measurement model. For example, using the best of the two worlds in order to obtain the best complexity/precision ratio.

Once we selected the estimation method, we detail computing power and algorithm complexity problematics in order to find available optimizations we can use to assess the usability of our system in a low power environment.

Then we present the developed methodology application to the UQS sensor, sold by TE Connectivity, study case. This sensor uses near infra-red spectroscopy to determine the urea concentration in a urea/water solution, in order to control the nitrogen-oxyde depolluting process in gasoline engines. After a design principles presentation, we detail the model we created in order to represent the system, to simulate its behavior and to combine the measurement data to extract the desired concentration. During this step, we focus on the obstacles of our model calibration and the deviation compensation, due to working conditions or to components aging process.

Based on this development, we finally designed the hybrid models addressing the nominal working cases and the model re-calibration during the working duration of the product. After this, we presented obtained results, on simulated data, and on real-world measured data. Finally, we enhanced the methodology based on tabulated “black box” models which are easier to calibrate and cheaper to process.

In conclusion, we reapplied our methodology to a different motion capture sensor, to compile all possible solutions and limits.



# Remerciements

Je tiens tout particulièrement à remercier le *LAAS-CNRS*, l'*INSA de Toulouse* et mes directeurs de thèse, Jean-Yves FOURNIOLS et Georges SOTO-ROMERO, ainsi que Pascal ACCO pour leur soutien, leur investissement et les enseignements dont ils m'ont gratifiés tout au long de cette thèse.

Je remercie également Jean MILPIED, Tom FOUGERE et la société *MEAS-France / TE Connectivity* pour m'avoir permis de réaliser ces travaux à leurs côtés, ainsi que pour leur souplesse.

Finalement, je remercie de tout coeur ma famille, mes amis et ma compagne pour leur amour et leur présence indéfectible dans les moments difficiles, et sans qui rien de tout cela n'aurait été possible.





# Table des matières

<b>Résumé</b>	<b>3</b>
<b>Abstract</b>	<b>5</b>
<b>Remerciements</b>	<b>7</b>
<b>Introduction générale</b>	<b>11</b>
<b>1 Etat de l'art</b>	<b>13</b>
1.1 Problématique du capteur intelligent . . . . .	13
1.2 Modélisation multiphysique et fusion de données . . . . .	15
1.2.1 Contexte et problématique . . . . .	15
1.2.2 La modélisation du système . . . . .	16
1.2.3 Exemple de modélisation et mesure indirecte linéaire : cas basique d'une sonde PT1000 . . . . .	19
1.2.4 Fusion de données pour une représentation plus globale de l'envi- ronnement . . . . .	23
1.2.5 Observation, fusion de systèmes non-linéaires . . . . .	26
1.2.6 Stratégie de calibration et représentation . . . . .	34
1.3 Conclusion . . . . .	36
1.3.1 Notre fil d'Ariane : le capteur UQS . . . . .	37
<b>2 Vers une sobriété technique, ou comment simplifier pour embarquer</b>	<b>41</b>
2.1 Un capteur au sein d'un système : la relation au temps . . . . .	41
2.1.1 Notions de temps réel . . . . .	42
2.1.2 Risques . . . . .	42
2.1.3 Verrous . . . . .	42
2.2 Quid des limitations techniques . . . . .	43
2.2.1 Limitations liées aux systèmes embarqués . . . . .	43
2.3 Complexité algorithmique . . . . .	44
2.3.1 La multiplication de matrice . . . . .	46
2.3.2 Addition de deux vecteurs . . . . .	46
2.3.3 Addition de deux matrices . . . . .	47
2.3.4 Transposition de matrice . . . . .	47
2.3.5 Inversion de matrice . . . . .	47
2.3.6 Calcul de vecteur moyen . . . . .	48
2.3.7 Calcul de matrice de covariance . . . . .	48
2.3.8 Récapitulatif . . . . .	49
2.4 Complexité du filtre de Kalman . . . . .	49

2.5	Diminution de la complexité des estimateurs . . . . .	50
2.5.1	Factorisation d'états . . . . .	51
2.5.2	Observateur à gain commuté . . . . .	51
2.5.3	Calculs en virgule fixe . . . . .	52
2.5.4	Pre-processing des mesures pour un système linéaire . . . . .	52
2.6	Conclusion . . . . .	52
<b>3</b>	<b>Un cas pratique industriel : le projet UQS</b>	<b>57</b>
3.1	Principe de fonctionnement . . . . .	57
3.1.1	Spectroscopie NIR . . . . .	58
3.1.2	Le capteur UQS . . . . .	59
3.1.3	Grandeurs d'entrées-sorties . . . . .	61
3.2	Paramètres d'influence et modèle . . . . .	62
3.2.1	La thermique au coeur du système . . . . .	62
3.2.2	Obstructions optiques . . . . .	62
3.2.3	Sensibilité mécanique . . . . .	63
3.3	Approche orientée modèle . . . . .	64
3.3.1	Paramètres de dispersion . . . . .	65
3.3.2	Paramètres de vieillissement . . . . .	65
3.3.3	Modèle des éléments optiques . . . . .	66
3.3.4	Modèle orienté signal du système optique . . . . .	69
3.3.5	Une dynamique thermique indépendante . . . . .	70
3.3.6	Modèle complet . . . . .	81
3.3.7	Estimation et stratégies de calibration . . . . .	81
3.3.8	Simulation . . . . .	87
3.3.9	Application sur des données capteur . . . . .	92
3.3.10	Question de puissance . . . . .	95
3.4	Approche sans modèle . . . . .	95
3.4.1	Méthode des LUTs de calibration . . . . .	96
3.5	Conclusion . . . . .	98
<b>4</b>	<b>Applicabilité de la méthode avec contrainte temps-réel</b>	<b>103</b>
4.1	Généralisation de la méthode . . . . .	104
4.2	Contraintes matérielles et temporelles . . . . .	104
4.3	Application de la méthode au système . . . . .	104
4.3.1	Essais en environnement contrôlé . . . . .	109
4.3.2	Utilisation pour l'analyse de la nage . . . . .	111
4.4	Conclusion . . . . .	111
	<b>Conclusion</b>	<b>117</b>
4.5	Histoire d'un aller et retour : application et réajustement de la méthode . .	117
4.6	Du diagnostique en temps réel . . . . .	118
4.7	Ouverture . . . . .	119

# Introduction générale

Doter un capteur d'une intelligence embarquée (*Smart sensors*) est un domaine en pleine expansion qui conjugue capteurs/conditionneurs et traitement de l'information au sens du diagnostic au plus près du capteur. Cette approche permet de conférer une adaptabilité du capteur à des variations de procédés de fabrication d'une part et à des variations d'environnement lors de l'utilisation du capteur en minimisant les procédés de recalibration pour compenser les dérives d'autre part. L'idée n'est pas nouvelle en soi mais l'avènement de microcontrôleurs de petite taille et à faible coût permet d'aller au-delà du conditionnement I2C ou SPI qu'on connaît pour contrôler *in situ* le signal en appliquant des techniques bayésiennes de filtrage pour corriger la mesure dans le capteur.

Issue de techniques maîtrisées pour des systèmes en capacité de calcul déportée, ce travail de recherche réalisé en partenariat avec la société *MEAS-France / TE Connectivity* et l'équipe S4M du *LAAS-CNRS* a pour objectif de mettre en place une méthodologie permettant la réalisation de capteurs embarqués intelligents capables d'estimer des paramètres physiques à partir de la fusion de données multi-physique. Cette stratégie devant permettre l'intégration de capteurs jusqu'alors réservés aux mesures en laboratoire dans des environnements défavorables, au sein de capteurs embarqués disposants de faibles puissances de calculs.

Après une exploration des méthodes liées à la modélisation, l'estimation de paramètres et les filtres de Kalman, nous détaillons différentes solutions existantes sur lesquelles nous appuyer pour répondre à la problématique de la fusion de données multi-physiques, pour les systèmes linéaires, avec le filtre de Kalman, les systèmes non-linéaires avec les filtres de *Extended Kalman Filter*, et *Unscented Kalman Filter*. Puis nous synthétisons un filtre pour les systèmes hybrides, présentant par exemple une évolution linéaire et une mesure non-linéaire, utilisant le meilleur des deux parties afin de tendre vers le meilleur compromis complexité/précision.

Une fois la méthode d'estimation sélectionnée, nous nous attardons sur les problématiques liées à la puissance de calcul et à la complexité algorithmique, afin de chercher les optimisations pouvant être mises en place afin de garantir le fonctionnement sur un système embarqué disposant d'une faible puissance de calcul.

Nous exposons ensuite l'application de cette méthodologie sur le cas d'étude du capteur UQS, commercialisé par *MEAS-France / TE Connectivity*, utilisant la spectroscopie proche infra-rouge pour déterminer la concentration d'urée en solution dans le cadre de la régulation des systèmes de dépollution des oxydes d'azote sur les moteurs diesel. Après une présentation des principes de fonctionnement, nous présentons le modèle développé que nous avons dressé afin de représenter le système, de simuler son comportement et d'effectuer la fusion de mesures permettant d'extraire la concentration désirée. Au cours de cette étape, nous abordons les problématiques liées à la calibration du modèle, ainsi qu'à la compensation de dérives, qu'elles soient liées aux conditions d'utilisation ou au vieillissement des composants. En s'appuyant sur cette réflexion, nous dressons finalement les

modèles hybrides nous permettant d'aborder les cas de fonctionnement nominaux, ainsi que les recalibrations du modèle au cours de la vie du produit. Après quoi, nous présentons les résultats obtenus, d'abord sur des données simulées, puis sur des données capturées sur un système réel, avant de proposer des pistes d'amélioration de la méthodologie en se basant sur des modèles tabulés en «boîte noire» qui permettent de faciliter l'étape de calibration tout en diminuant la charge de calcul.

Finalement, nous généralisons la méthodologie par le biais d'une problématique de capture de mouvement, pour conclure sur les possibilités et les limites de la solution que nous avons proposée.

Ce manuscrit est organisé en quatre chapitres :

- Le premier chapitre présente un état de l'art, d'une part autour des capteurs intelligents, d'autre part autour des méthodologies de fusion de données par filtrage bayésien en s'appuyant sur un modèle de système,
- dans le second chapitre nous explorons les questions liées à la complexité algorithmique et à l'optimisation des algorithmes dans le but de les embarquer sur micro-contrôleur pour effectuer du traitement en temps-réel,
- le troisième chapitre applique la méthodologie présentée au capteur industriel UQS, en détaillant le modèle dressé du capteur, en y appliquant un filtre de Kalman et en exploitant les résultats,
- Le dernier chapitre est dédié à la généralisation de la méthode illustrée par un capteur de capture de mouvement.

Ce manuscrit se termine par une conclusion générale résumant les différents points et travaux discutés et présentant les perspectives tracées pour la suite de ce projet.

# Chapitre 1

## Etat de l'art

### 1.1 Problématique du capteur intelligent

Les stratégies de traitement du signal sont au cœur de la performance métrologique de tout capteur dans son environnement. Dans le cadre de ses activités de développement de capteurs embarqués, *MEAS-France / TE Connectivity* souhaite définir et mettre en œuvre des algorithmes innovants de traitement numérique du signal d'un capteur de gaz embarqué. A dimension applicative immédiate, ces travaux de thèse ont été conduits en collaboration entre le site *MEAS-France / TE Connectivity France* de Toulouse et le Laboratoire d'Analyse et d'Architecture des Systèmes (*LAAS*) l'équipe S4M. Destinés à être installés sur une plateforme hardware existante, les travaux conduits ont pour objectifs scientifiques principaux la définition, le codage et la validation de méthodes algorithmiques innovantes et intelligentes permettant de stabiliser et d'optimiser les performances métrologiques d'un capteur en lui associant une intelligence embarquée.

Le contexte industriel de ces travaux concerne la problématique de respect des normes environnementales et particulièrement des normes antipollution pour l'automobile, actuellement Euro6 qui pousse les constructeurs de véhicules particuliers et de poids lourds vers la mise en œuvre de technologies avancées de contrôle des émissions. Les systèmes catalytiques de post-traitement des gaz d'échappement jouent un rôle principal dans le contrôle embarqué de ces émissions (HC, CO, NOx, particules)[1]. Citons le système catalyseur 3 voies pour le moteur à allumage commandé par exemple, ou encore, le système SCR (Selective Catalysis Reduction) pour le moteur Diesel. Etant donné l'engagement au respect de ces normes durant la vie du véhicule (>300 000km) les constructeurs et équipementiers sont tenus d'assurer le bon fonctionnement de ces systèmes catalytiques durant leur opération et surtout à alerter le système de contrôle moteur en cas de dysfonctionnement : c'est l'On Board Diagnosis ou OBD. La norme OBD implique la mise en œuvre de nombreux capteurs embarqués sur les systèmes d'échappement pour contrôler l'opérabilité et la performance des systèmes de dépollution et donc le bon respect des normes d'émissions[2]. Cependant, des verrous persistent tels que la non linéarité de la réponse, la dérive lente à long terme ou l'influence de certaines conditions environnementales sur la sensibilité, telles que la température des gaz. Au-delà de la robustesse et du design de la technologie de mesure, l'optimisation des performances du capteur ne peut se concrétiser que par l'intégration d'une gestion intelligente de la mesure et donc par la mise en œuvre de stratégies algorithmiques de traitement du signal [3][4].

Du point de vue académique, la thématique du smart sensing, ou plus exactement pour nous du «smart sensor», est d'actualité avec près de 10M citations sous un moteur

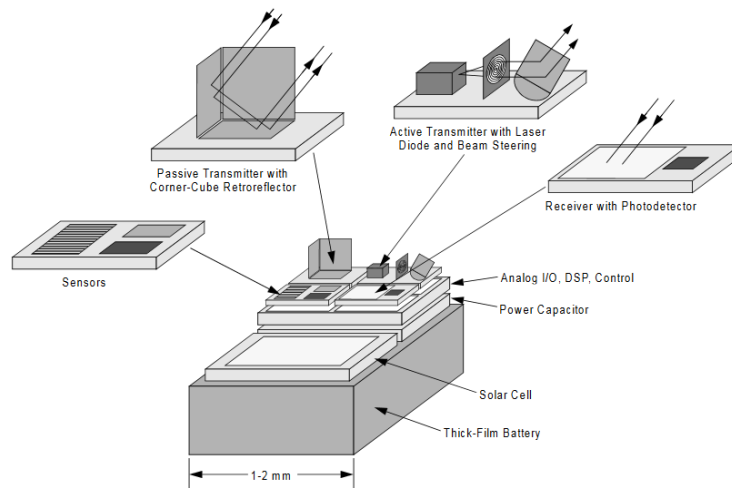


FIGURE 1.1 – SmartDust [5]

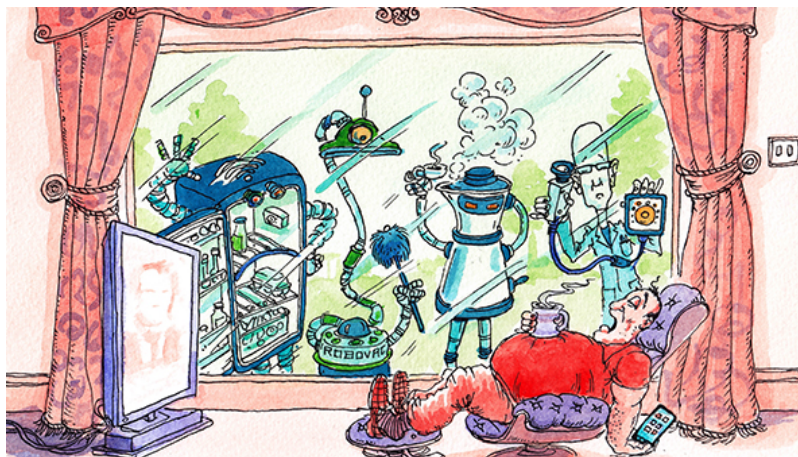


FIGURE 1.2 – The Internet of Things [6]

de recherche classique. Le monde est SMART y compris dans de marketing télécom qui prône la techno attitude...

Permettons-nous à notre niveau de définir notre positionnement dans le «smart sensor» en s'appuyant sur deux publications qui semblent différentes mais que nous tentons d'agrèger. Quel est le point commun entre la publication originelle de Kalman en 1960 [7] et la publication «Smart Dust» de Berkeley en 1999 ? Pour la première c'est une technique connue et maîtrisée ; pour la seconde c'est l'avènement du marché de l'IoT.

Est-il possible d'associer localement à un capteur une approche «filtrage de Kalman» avec les contraintes de l'embarqué (prix, consommation électrique) pour conférer au nœud de mesure une adaptabilité et réduire la quantité de datas transmises «dans le cloud» opéré par les «data scientists» ? Différents travaux tendent à partager cette vision d'architectures versatiles au plus près du capteur afin de conférer un niveau de «robustesse» à la mesure et ainsi, soit s'affranchir des variations environnementales, soit même penser à «compenser» par le filtrage des variations de fabrication et ainsi modifier la façon de penser la technologie ultra reproductible puisque le filtrage pourrait compenser ces variations de comportement.

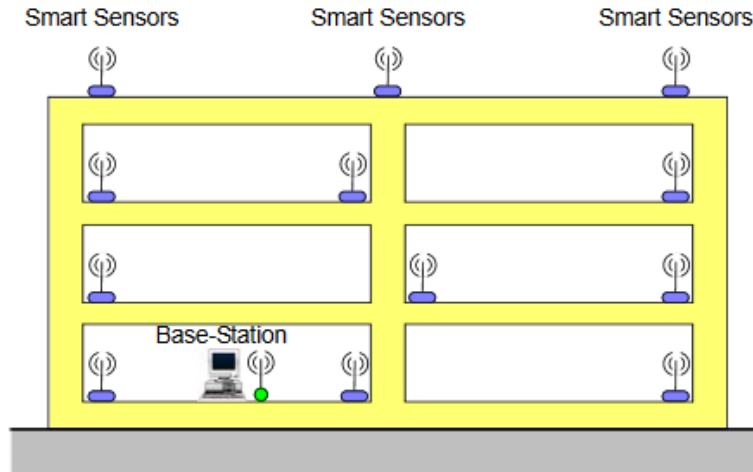


FIGURE 1.3 – Smart sensors [8]

## 1.2 Modélisation multiphysique et fusion de données

### 1.2.1 Contexte et problématique

Avec l'évolution des technologies, les objets connectés ou le durcissement des normes anti-pollution, le besoin de capteurs embarqués n'a jamais été aussi important qu'à notre époque. Les besoins vont de l'équipement de loisir (domotique, mesure de données physiologiques pour les sports...) aux applications critiques (détecteur de fumée, de monoxyde de carbone, de collision pour les véhicules...) en passant par des applications industrielles (contrôle de qualité en production, automatisation des procédés...).

Cette augmentation de la demande va de pair avec un besoin de diminution des coûts de production, d'accélération des temps de développement, de facilitation de l'utilisation (en retirant les procédures de calibration avant chaque utilisation par exemple), d'augmentation de la précision et de diminution des temps de réponse. A cela, nous pouvons ajouter le besoin de miniaturisation des capteurs, qui est un axe favorisant les facteurs précédemment cités.

Jusqu'à présent, la réalisation de capteurs consistait à mettre en place des designs spécifiques s'appuyant sur les propriétés physiques de matériaux souvent coûteux. Certains capteurs devaient de plus être ré-étalonnés dans un échantillon de référence avant chaque utilisation (cas du Ph-mètre), ou fonctionner dans des conditions contrôlées (température, hygrométrie...) afin de garantir la précision maximale. Cela impose des procédés volumineux et/ou contraignants à utiliser, qui ne sont pas applicables dans le cas de mesure embarquée en continue.

L'une des solutions pour adresser simultanément tous ces besoins consiste à assembler un ensemble de capteurs multi-physique disponibles «sur étagère» afin de remplir la tâche désirée.

L'augmentation des plages de mesure, le non contrôle de l'environnement (qui est inhérent à la notion de capteur embarqué) et l'utilisation d'éléments non dédiés, sensibles à plusieurs grandeurs physiques autres que celle que nous souhaitons mesurer augmentent la difficulté de traitement de l'information. En effet, il n'est dans ces conditions plus possible d'observer directement la grandeur physique mesurée par simple mesure de l'un des éléments constitutifs du capteur.



Il sera donc nécessaire d'effectuer un traitement intelligent, ou une fusion de données, des différentes informations fournies par les éléments constitutifs du capteur afin de s'affranchir des grandeurs d'influences non désirées et de restituer une mesure précise et peu dépendante des conditions environnementales.

## 1.2.2 La modélisation du système

La modélisation du système consiste à déterminer la fonction permettant, à partir d'un certain nombre d'entrées et de paramètres d'influence de prédire les mesures que nous observerons en sortie.

Dans la suite de cette section, nous utiliserons indistinctement le mot système pour désigner soit le capteur complet, soit l'un de ses éléments constitutifs.

Dans le cas de systèmes composés, tels que les capteurs que nous nous proposons d'étudier, cette modélisation passe d'abord par la modélisation de ses éléments constitutifs, puis par l'intégration du système complet.

Pour cela, il faut avant tout dresser la liste des entrées-sorties, ainsi que des paramètres d'influence du système.

Une fois cette liste dressée, il est possible de diviser le comportement du système selon deux axes : le comportement statique et le comportement dynamique.

Le premier représente le comportement du système lorsque ses entrées, son état interne et ses sorties sont stables. Il s'agit généralement du comportement utile pour la mesure de l'information. Il peut être représenté par une relation entrées/paramètres/sorties indépendante du temps. On peut ainsi obtenir l'équation (1.1) pour représenter les mesures  $\mathbf{s}$  disponibles en sortie du système pour une combinaison d'entrées  $\mathbf{e}$  et de paramètres  $\mathbf{\Gamma}$ .

$$\mathbf{s} = h(\mathbf{e}, \mathbf{\Gamma}) \quad (1.1)$$

Ce modèle statique peut être extrait depuis les modèles physiques ou les données constructeur des composants utilisés. Lorsque ces données ne sont pas disponibles, il est nécessaire d'effectuer une caractérisation, ce qui consiste en une série de mesures réalisées sur un ensemble d'échantillons de systèmes, en faisant varier indépendamment puis de façon croisée, chacun des paramètres d'influence et chacune des entrées du système. Le modèle statique du système peut alors être dressé par identification de fonctions mathématiques multi-variables ou, lorsque le système est fortement non-linéaire et ne peut être identifié de façon simple, par des tableaux de données multi-dimensionnelles (une dimension par entrée, grandeur d'influence et sortie du système). Cette seconde méthode est appelée tabulation (*Look-Up Table*, LUT en anglais).

Si l'échantillon est suffisamment important, cette caractérisation pourra nous donner le modèle statique du système mais aussi un modèle de dispersion qui pourra être utilisé en phase de calibration pour faire converger le modèle générique vers le système réel.

En répétant ce même test de façon régulière sur un temps suffisamment long, il sera aussi possible de dresser un modèle statistique d'évolution des paramètres internes du système lié à son vieillissement.

Le **modèle dynamique** quant à lui, représente le comportement du système lors des phases transitoires (lorsque l'une ou plusieurs de ses entrées, de ses paramètres d'influence ou de ses états internes varient dans le temps).

On peut alors distinguer deux fonctions :

- l'évolution de l'état interne du système (que nous noterons  $\mathbf{x}(t)$ ) en fonction de l'évolution des entrées et des paramètres d'influence :

$$\dot{\mathbf{x}}(t) = f(\mathbf{x}(t), \mathbf{e}(t), \mathbf{\Gamma}(t)) \quad (1.2)$$

— l'évolution des mesures en sortie du système en fonction de l'état interne, des entrées et de paramètres :

$$\mathbf{s}(t) = h(\mathbf{x}(t), \mathbf{e}(t), \mathbf{\Gamma}(t)) \quad (1.3)$$

La première équation étant appelée équation d'évolution du système et représentant la dynamique du système, la seconde étant appelée équation de mesure et représentant la partie statique entre l'état interne et la sortie du système. Ces deux fonctions peuvent être influencées par un bruit (appelé bruit d'état pour la première et bruit de mesure pour la seconde).

Ainsi, une LED a pour grandeur d'entrée un courant et pour valeur de sortie un spectre lumineux d'intensité plus ou moins importante. Cependant, cet élément est aussi sensible à la température et peut voir aussi bien son spectre que sa puissance d'émission varier fortement en fonction de ce paramètre.

Le comportement statique de la LED peut être caractérisé en faisant varier le courant d'entrée  $i(t)$  et la température  $T(t)$  et en mesurant l'émission  $s_{spectre}(t)$  à l'aide d'un spectroscope en régime stable.

Son comportement dynamique peut se décomposer à 2 échelles temporelles :

- Sur une très petite période : le comportement de l'émission en fonction d'une variation de courant ou de température,
- Sur une longue période : la dégradation des performances de la LED due à son vieillissement.

Nous avons les équations d'état :

$$\dot{i} = f_i(v(t), T_{led}(t)) \quad (1.4a)$$

$$\dot{T}_{led}(t) = f_{T_{led}}(T_{led}, T_{amb}) \quad (1.4b)$$

Nous avons donc, pour une LED la fonction de sortie (1.5).

$$s_{spectre}(t) = h_{led}(i(t), T_{led}(t)) \quad (1.5)$$

Une fois tous les éléments constitutifs modélisés, il est possible de modéliser le capteur complet comme un assemblage de ces éléments. Il peut alors être nécessaire d'effectuer des caractérisations complémentaires afin de déterminer les interactions entre certains éléments (par exemple l'échauffement d'un composant pouvant modifier le point de fonctionnement d'un autre).

Lorsque les différentes caractéristiques ont été déterminées, il est alors possible de représenter le système sous la forme d'un modèle afin d'en déterminer les propriétés et de pouvoir en extraire les mesures utiles à notre objectif de capteur intelligent. Pour représenter ce modèle, il existe deux grandes catégories que nous allons détailler ci-dessous :

- le modèle linéaire invariant dans le temps,
- le modèle non-linéaire.

## Modèle Linéaire Invariant dans le Temps

Un système Linéaire Invariant dans le Temps (LIT) peut être défini selon le chapitre I de [9].

Dans le cas d'un capteur, cela suppose que :

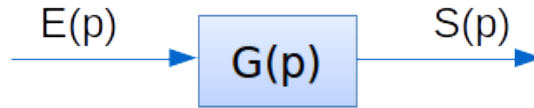


FIGURE 1.4 – Fonction de transfert  $G(p)$  d'un système

- le capteur ait une réponse linéaire ( $g(\mathbf{e} + \Delta\mathbf{e}) = g(\mathbf{e}) + g(\Delta\mathbf{e})$ )
- le capteur ne manifeste aucune notion de vieillissement (modification lente de paramètres du modèle au cours de la vie du capteur).

Les propriétés de tels systèmes ont été largement étudiées par le passé [10] [9] et présentent de forts avantages quant à l'analyse du système.

Ainsi, chaque système LIT peut être représenté comme une combinaison linéaire de la contribution de chacune de ses entrées (théorème de superposition). Chaque contribution peut donc être étudiée séparément de façon décorrélée lorsque les hypothèses de LIT sont respectées, ce qui simplifie grandement l'étape de modélisation.

De tels systèmes répondent généralement à un jeu d'équations différentielles qui régissent leur évolution. Il est alors commode d'utiliser la transformée de Laplace pour analyser les comportements associés. Ceux-ci sont alors vus comme des combinaisons linéaires dans le domaine de Laplace ( $p$  étant la variable de Laplace,  $F(p)$  est appelée fonction de transfert du système, *cf.* 1.2.2) (1.6) :

$$S(p) = \int_{0+}^{+\infty} s(t) \exp(-pt) dt = \int_{0+}^{+\infty} g(e(t)) \exp(-pt) dt = G(p)E(p) \quad (1.6)$$

D'un point de vue pratique, ces modèles représentent parfaitement des systèmes linéaires tels que de l'électronique linéaire (résistances, condensateurs, inductances, amplificateurs) ou des comportements physiques (thermique, diffusion...). Ils ne sont cependant pas applicables dans le cas de composants présentant des non-linéarités (diodes, CTN...), à moins d'effectuer une linéarisation autour de points de fonctionnement.

Le système de représentation le plus simple utilisé pour les systèmes mono-entrée, mono-sortie est la fonction de transfert. Ce système réagit donc selon l'équation (1.6). La fonction de transfert de ce système est alors définie par l'équation (1.7).

$$\frac{S(p)}{E(p)} = G(p) \quad (1.7)$$

$G(p)$  étant la fonction de transfert du système dans le domaine de Laplace. On peut alors représenter le système selon la Figure 1.4.

### Modèle non linéaire

Pour les systèmes présentant des non-linéarités, les hypothèses précédentes ne sont plus valides. Il est alors impossible de décrire le comportement du système sous la forme de combinaisons linéaires des contributions des différentes entrées.

De même, la décorrélation des entrées les unes par rapport aux autres n'est plus garantie, ce qui nécessite une caractérisation plus poussée en croisant différents paramètres pour couvrir l'ensemble de l'espace d'états auquel le système peut être soumis (voir 1.2.2).

Cette modélisation est plus générale, mais plus difficile à traiter. Ainsi, les évolutions du système font souvent intervenir des équations différentielle non linéaires et les solutions des intégrales multidimensionnelles.

Afin de simplifier l'analyse de tels systèmes, lorsque les non-linéarités sont suffisamment faibles, il est fréquent de les étudier comme systèmes LIT en effectuant une linéarisation locale autour d'un certain nombre de points de fonctionnement. Nous retrouvons alors les propriétés liées aux systèmes LIT [9].

Une fois les propriétés du modèle établies, il est important de fixer une convention de représentation afin de l'étudier plus en détail. Nous allons détailler deux types de représentations communes en automatique dans la section suivante.

### 1.2.3 Exemple de modélisation et mesure indirecte linéaire : cas basique d'une sonde PT1000

Dans cette section nous allons aborder deux des principales représentations de systèmes utilisées en automatique : la fonction de transfert et la représentation d'états.

Ces deux conventions présentent chacune leurs avantages et inconvénients que nous allons tenter d'exposer ici. Dans les deux cas, nous supposons que le système est LIT ou linéarisé autour d'un point de fonctionnement.

Cette représentation présente un intérêt lors de la réalisation de systèmes composés. Ainsi, par exemple, nous pouvons représenter le système multi-physique linéaire simple représenté en Figure 1.5 sous la forme d'un assemblage de fonctions de transfert.

Nous l'utilisons pour illustrer notre démarche de modélisation adaptée à la fusion de données de capteurs.

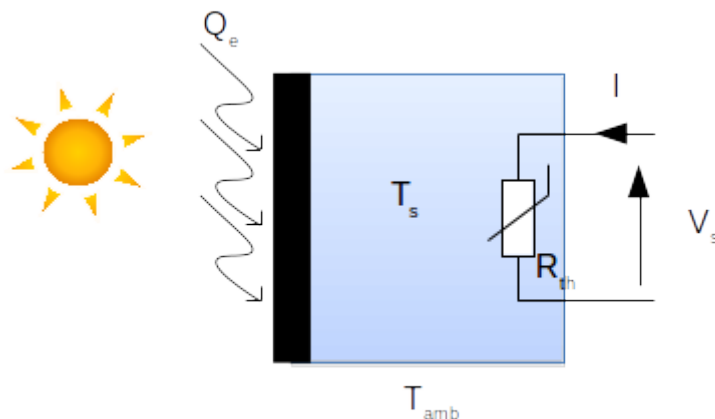


FIGURE 1.5 – Système multiphysique linéaire simple

Le système proposé ici est un capteur de rayonnement solaire basé sur un corps thermique de capacité thermique  $C_s$  et de résistance thermique  $R_s$ , déterminés par les propriétés du matériau et la géométrie du système, vis-à-vis de la température ambiante  $T_{amb}$ . La température interne du système est désignée par la grandeur  $T_s$ , exprimée en  $^{\circ}\text{C}$  et mesurée via la résistance PT1000  $R_{th}$ , alimentée par un courant constant  $I$ .

Nous supposons ici que la température ambiante  $T_{amb}$  est constante et connue.

L'entrée du système est désignée par le flux thermique  $Q_e(t)$  et la sortie du système est la tension  $V_s(t)$ .

Le système peut être séparé en deux parties :

- le comportement thermique,
- la conversion thermo-électrique.

Le comportement thermique répond à l'équation (1.8).

$$Q_e(t) - \frac{T_s(t) - T_{amb}}{R_s} = C_s \frac{\delta T_s(t)}{\delta t} = C_s \dot{T}_s(t) \quad (1.8)$$

Avec  $\frac{\delta T_s(t)}{\delta t} = \dot{T}_s(t)$  la dérivée partielle de  $T_s(t)$  en fonction du temps. Ce qui, en transformée de Laplace donne :

$$T_s(p) = \frac{R_s}{1 + R_s C_s p} Q_e(p) + T_{amb}(p) \quad (1.9)$$

Cette partie peut donc se représenter selon la Figure 1.6.

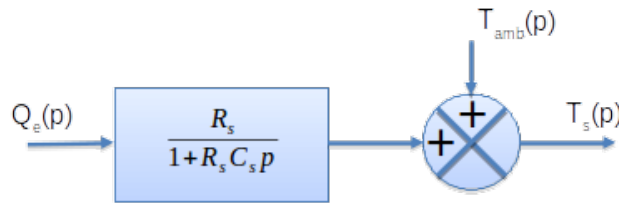


FIGURE 1.6 – Schéma bloc du comportement thermique

Pour poursuivre le schéma bloc du comportement de la PT1000, on prend en compte la conversion thermo-électrique selon les équations (1.10a,1.10b).

$$R_{th}(t) = R_0 \left(1 - \frac{T_s(t)}{1000}\right) \quad (1.10a)$$

$$V_s(t) = R_{th}(t)I = IR_0 - \frac{IR_0}{1000} T_s(t) \quad (1.10b)$$

On en déduit le schéma bloc de la conversion (Figure 1.7) que l'on vient associer au système précédent pour définir le système complet(Figure 1.8).

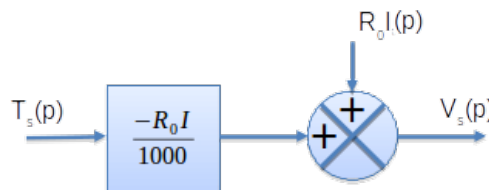


FIGURE 1.7 – Schéma bloc de la conversion thermo-électrique

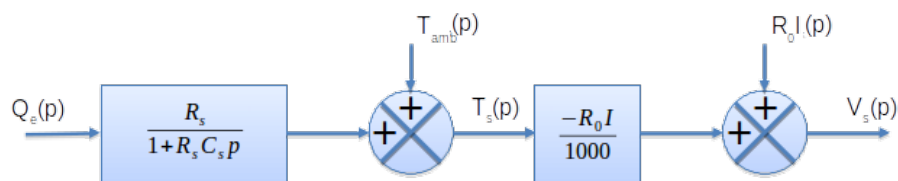


FIGURE 1.8 – Schéma bloc du système complet

Il est ainsi possible de calculer simplement les fonctions de transfert du système :

$$G_{s/Q_e}(p) = \frac{V_s(p)}{Q_e(p)} = \frac{R_s}{1 + R_s C_s p} \left( \frac{-R_0 I}{1000} \right) = -\frac{R_0 R_s I}{1000(1 + R_s C_s p)} \quad (1.11a)$$

$$G_{s/T_{amb}}(p) = \frac{V_s(p)}{T_{amb}(p)} = \frac{-R_0 I}{1000} \quad (1.11b)$$

$$G_{s/R_0 I}(p) = \frac{V_s(p)}{R_0 I(p)} = 1 \quad (1.11c)$$

Nous remarquons ainsi que même dans le cas d'un système simple, le nombre de fonctions de transfert à calculer devient important. De plus, la notion d'évolution temporelle du système n'est pas parfaitement explicitée par cette représentation,  $G(p)$  étant une relation d'entrées/sorties qui masque l'état du système (et donc l'information que nous souhaitons observer). Pour ces raisons, la représentation d'état est généralement préférée dès que la complexité du système devient importante.

### La représentation d'état

A partir du moment où la complexité d'un système ne peut plus se résumer à une simple relation d'entrées/sorties, ou quand il est nécessaire de représenter des paramètres et/ou états internes du système, la représentation d'état devient la solution privilégiée.

Ainsi, nous définissons les  $n$  états internes de notre système par le vecteur  $\mathbf{X}$  (appelé vecteur d'états) tel que :

$$\mathbf{X} = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} \quad (1.12)$$

Les  $m$  entrées du système sont définies par le vecteur  $\mathbf{U}$ , appelé vecteur de commande :

$$\mathbf{U} = \begin{pmatrix} u_1 \\ \vdots \\ u_m \end{pmatrix} \quad (1.13)$$

Et les  $k$  sorties sont représentées par le vecteur  $\mathbf{Y}$ , appelé vecteur de mesure :

$$\mathbf{Y} = \begin{pmatrix} y_1 \\ \vdots \\ y_k \end{pmatrix} \quad (1.14)$$

Ainsi, le système général est régi par les équations :

$$\dot{\mathbf{X}}(t) = f(\mathbf{X}(t), \mathbf{U}(t)) \quad (1.15a)$$

$$\mathbf{Y}(t) = h(\mathbf{X}(t), \mathbf{U}(t)) \quad (1.15b)$$

$\mathbf{X}$  et  $\mathbf{Y}$  pouvant être affectés par des bruits.

La force de cette convention est de représenter l'évolution des états internes (états réels, paramètres de calibration, entrées non commandées) du système, ainsi que toutes leurs

interactions. Mais il présente un avantage encore plus important lorsque nous considérons le système comme étant LIT.

En effet, dans ce cas les équations d'évolution et de mesure deviennent :

$$\dot{\mathbf{X}}(t) = [\mathbf{A}]\mathbf{X}(t) + [\mathbf{B}]\mathbf{U}(t) \quad (1.16a)$$

$$\mathbf{Y}(t) = [\mathbf{C}]\mathbf{X}(t) + [\mathbf{D}]\mathbf{U}(t) \quad (1.16b)$$

Avec

- $[\mathbf{A}]$  la matrice d'évolution de dimensions  $[n, n]$ ,
- $[\mathbf{B}]$  la matrice de commande de dimensions  $[n, m]$ ,
- $[\mathbf{C}]$  la matrice de mesure de dimensions  $[k, n]$ ,
- $[\mathbf{D}]$  la matrice direct de dimensions  $[k, m]$  (cette matrice est généralement nulle).

En dérivant un modèle d'état linéaire, il devient possible par analyse des propriétés de  $[\mathbf{A}]$ ,  $[\mathbf{B}]$ ,  $[\mathbf{C}]$  et  $[\mathbf{D}]$  de déterminer des propriétés importantes telles que la commandabilité et l'observabilité des états  $\mathbf{X}$  du système.

Pour notre exemple, nous pouvons placer la mesure  $Q_e$  dans l'état du système. En effet, il ne s'agit pas d'une entrée du système au sens où nous ne la connaissons pas a priori et nous cherchons à estimer sa valeur. Ce capteur peut alors se représenter par les vecteurs d'état, de commande et de mesure suivants :

$$\mathbf{X} = \begin{pmatrix} Q_e \\ T_s \end{pmatrix} \quad (1.17a)$$

$$\mathbf{Y} = \begin{pmatrix} V_s \end{pmatrix} \quad (1.17b)$$

$$\mathbf{U} = \begin{pmatrix} T_{amb} \\ 1 \end{pmatrix} \quad (1.17c)$$

Et les matrices suivantes :

$$[\mathbf{A}] = \begin{bmatrix} 0 & 0 \\ \frac{1}{C_s} & \frac{-1}{R_s C_s} \end{bmatrix} \quad (1.18a)$$

$$[\mathbf{B}] = \begin{bmatrix} 0 & 0 \\ \frac{1}{R_s C_s} & 0 \end{bmatrix} \quad (1.18b)$$

$$[\mathbf{C}] = \begin{bmatrix} 0 & \frac{-R_0 I}{1000} \end{bmatrix} \quad (1.18c)$$

$$[\mathbf{D}] = \begin{bmatrix} 0 & R_0 I \end{bmatrix} \quad (1.18d)$$

## Observabilité

D'après [9], l'observabilité d'un système est définie par :

Un système est dit observable à un instant  $t_1$ , si la connaissance du signal d'entrée et du signal de sortie sur l'intervalle de temps  $[t_1, t_2]$  permet de calculer l'état du système à l'instant  $t_2$ .

Dans le cas qui nous intéresse, cela signifie que si notre capteur est observable, il est possible d'estimer son état interne, dans lequel nous aurons au préalable mis la mesure qui nous intéresse. Il s'agit donc d'une condition nécessaire pour nous permettre de réaliser la fusion de données et en extraire la grandeur physique désirée de façon non biaisée.

Ce critère est déterminé pour un système linéaire par le calcul du rang de la matrice d'observabilité  $[\mathbf{O}]$  définie par (1.19) ( $n$  étant le nombre d'états du système).

$$[\mathbf{O}] = \begin{bmatrix} \mathbf{C} \\ \mathbf{C}[\mathbf{A}] \\ \vdots \\ \mathbf{C}[\mathbf{A}]^{n-1} \end{bmatrix} \quad (1.19)$$

Il faut que la matrice  $[\mathbf{O}]$  soit régulière (donc  $\det([\mathbf{O}]) \neq 0$ ).

Dans le cas de notre capteur de rayonnement solaire, nous obtenons :

$$[\mathbf{O}] = \begin{bmatrix} 0 & \frac{-R_0 I}{1000} \\ \frac{-R_0 I}{1000 C_s} & \frac{R_0 I}{1000 R_s C_s} \end{bmatrix} \quad (1.20)$$

$$\det([\mathbf{O}]) = -\frac{R_0^2 I^2}{10^6 C_s} \quad (1.21)$$

Ainsi, le flux thermique  $Q_e$  est observable si  $R_0 \neq 0$  et  $I \neq 0$  sur ce capteur.

Il devient alors possible de vérifier si le système est observable, et dans le cas contraire, de vérifier si l'ajout de points de mesures (augmentation de  $\mathbf{Y}$ , et donc de  $[\mathbf{C}]$ ) permet de le rendre observable. Par exemple, on voit ici qu'il est inutile de mesurer  $I$  en plus de  $V$  pour observer  $Q_e$ .

Une fois l'observabilité obtenue sur notre système (ou modèle), il s'agit de mettre en place un observateur, ou un estimateur, permettant d'estimer l'état du modèle à partir des mesures. Dans notre cas, nous confions cette tâche à un filtre de Kalman, ce qui est l'une des solutions les plus étudiées pour effectuer de la fusion de données multi-capteurs [11][12].

### 1.2.4 Fusion de données pour une représentation plus globale de l'environnement

Dans le cas étudié ici, nous réalisons la fusion de données (entrées/sorties de notre système de mesure) à l'aide d'un observateur, dans le but d'avoir une estimation la plus juste possible de l'état de notre système (contenant les paramètres, états internes ainsi que la (ou les) grandeur(s) physique(s) que nous souhaitons mesurer, *cf.* capteur de rayonnement solaire 1.2.3).

#### Les observateurs

Le principe de base d'un observateur consiste à estimer l'état d'un système observable à partir de ses entrées et de ses sorties. Pour ce faire, la solution consiste à réaliser le précédé en trois étapes :

- Estimer l'état du système depuis le modèle :

$$\dot{\hat{\mathbf{X}}}(t) = f(\hat{\mathbf{X}}(t), \mathbf{U}(t)),$$



— Prédire les sorties observées :

$$\tilde{\mathbf{Y}}(t) = h(\tilde{\mathbf{X}}(t), \mathbf{U}(t)),$$

— Corriger l'état par propagation de l'erreur d'estimation à partir de la mesure  $\mathbf{Y}(t)$  :

$$\hat{\mathbf{X}}(t) = \tilde{\mathbf{X}}(t) + [\mathbf{K}](\mathbf{Y}(t) - \tilde{\mathbf{Y}}(t))$$

Différent types d'observateurs peuvent être réglés en modifiant la valeur de  $[\mathbf{K}]$ . Le cas optimal en terme d'erreur quadratique est l'estimateur Bayésien ([13]) dans le cadre des systèmes continus en temps et en état. Ce filtre se base sur un calcul de la densité de probabilité de l'espace d'états et des mesures pour propager à chaque instant l'erreur de mesure proportionnellement à cette probabilité. Le gain de correction  $[\mathbf{K}]$  est ainsi mis à jour de façon continue en fonction de l'état courant estimé du système. Cet estimateur représente cependant une charge de calcul trop importante pour être réalisé par un calculateur sur un système dont l'espace d'état est trop grand. Nous allons donc nous concentrer sur une simplification du filtre Bayésien sous certaines hypothèses.

## La discrétisation

Le matériel nous permettant d'effectuer l'estimation de l'état de notre système étant une unité de calculs informatique (ordinateur, microcontrôleur...), il est incapable, par nature, de travailler sur des signaux continus tels que décrits précédemment. Ces derniers doivent être transformés en une suite de nombres pour pouvoir être traités [9]. Il s'agit de la discrétisation.

Toutes les méthodes présentées ici s'appuieront sur des données et des modèles de systèmes discrétisés.

Ce changement a un impact au niveau de la représentation du système. Ainsi nous ne calculons plus la dérivée de l'état du système mais la valeur de celui-ci à l'instant suivant, que nous noterons :

—  $\mathbf{X}_k$  l'état à l'instant  $t_k = k T_e$ ,

—  $\mathbf{X}_{k+1}$  l'état à l'instant suivant  $t_{k+1} = (k + 1) T_e$ ,

Pour une transformation d'un système à temps continu vers un système à temps discret, échantillonné avec une période d'échantillonnage  $T_e$ , nous utilisons la solution générale de l'équation d'évolution pour un état initial  $\mathbf{X}_0 = \mathbf{X}(t_0)$  et une entrée  $u(t)$  :

$$\mathbf{X}(t) = e^{[\mathbf{A}](t-t_0)} \mathbf{X}_0 + \int_{t_0}^t e^{[\mathbf{A}](t-\tau)} [\mathbf{B}] \mathbf{u}(\tau) d\tau \quad (1.22)$$

Comme la commande  $u(t)$  est constante pendant chaque période d'échantillonnage, modélisé par un bloqueur d'ordre zéro, et en prenant  $\mathbf{X}_k$  comme condition initiale de l'équation précédente il vient :

$$\mathbf{X}_{k+1} = e^{[\mathbf{A}]T_e} \mathbf{X}_k + \left\{ \int_0^{T_e} e^{[\mathbf{A}]\tau} d\tau \right\} [\mathbf{B}] \mathbf{u}_k \quad (1.23)$$

En prenant la forme développée de l'exponentielle d'une matrice  $e^{[\mathbf{A}]t} = \sum_{n=0}^{\infty} [\mathbf{A}]^n \frac{t^n}{n!}$ , l'intégrale précédente se simplifie et l'expression devient :

$$\mathbf{X}_{k+1} = e^{[\mathbf{A}]T_e} \mathbf{X}_k + [\mathbf{A}]^{-1} (e^{[\mathbf{A}]T_e} - \mathbf{I}) [\mathbf{B}] \mathbf{u}_k \quad (1.24)$$

L'équation de sortie étant une combinaison linéaire valable à tout instant, elle reste inchangée. Nous obtenons donc un modèle d'état discret représentant exactement le comportement du système continu échantillonné–bloqué par la forme :

$$\mathbf{X}_{k+1} = e^{[\mathbf{A}]T_e} \mathbf{X}_k + [\mathbf{A}]^{-1} (e^{[\mathbf{A}]T_e} - \mathbf{I}) [\mathbf{B}] \mathbf{u}_k \quad (1.25)$$

$$\mathbf{Y}_k = [\mathbf{C}]\mathbf{X}_k + [\mathbf{D}] \mathbf{U}_k \quad (1.26)$$

Cette représentation reste exacte quelle que soit la période d'échantillonnage choisie. Cependant dans le cas d'une modélisation analytique, le calcul de l'exponentielle et de l'inverse matricielle peut donner des résultats trop complexes à utiliser ou tout simplement être trop complexe à calculer en un temps raisonnable. Dans ce cas, des transformations approchées demandant des fréquences d'échantillonnage plus élevées peuvent être utilisées : ainsi l'intégrale 1.22 peut être simplifiée en utilisant le développement partiel de l'exponentielle  $e^{AT_e} \approx \mathbf{I} + AT_e$  ce qui permet d'éviter le calcul de l'inverse et donne :

$$\mathbf{X}_{k+1} = e^{[\mathbf{A}]T_e} \mathbf{X}_k + T_e [\mathbf{B}] \mathbf{u}_k \quad (1.27)$$

$$\mathbf{Y}_k = [\mathbf{C}]\mathbf{X}_k + [\mathbf{D}] \mathbf{u}_k \quad (1.28)$$

En remplaçant aussi le terme exponentiel restant par son développement partiel, la solution se réduit à une méthode d'intégration considérant que la dérivée de l'état est constante et égale à  $\dot{\mathbf{X}}(t_k) = \mathbf{A} X(t_k) + [\mathbf{B}] u(t_k)$  pendant une période d'intégration. On obtient alors dans ce cas la forme suivante :

$$\mathbf{X}_{k+1} = \mathbf{X}_k + T_e [\mathbf{A}] \mathbf{X}_k + T_e [\mathbf{B}] \mathbf{u}_k \quad (1.29)$$

$$\mathbf{Y}_k = [\mathbf{C}]\mathbf{X}_k + [\mathbf{D}] \mathbf{u}_k \quad (1.30)$$

## Le filtre de Kalman (KF)

Le filtre de Kalman [7] est une simplification du filtre Bayésien pour un système linéaire à temps discret pour lequel les bruits de mesure et d'état sont supposés gaussiens.

Ces deux hypothèses fortes permettent de réduire la complexité de l'estimateur à un calcul de moyennes et de covariances au travers de transformations linéaires.

En plus des vecteurs d'état, de mesure et de commande précédemment cités, nous ajoutons ici à nos équations d'évolution et de mesure des vecteurs de bruits (respectivement  $\mathbf{V}$  et  $\mathbf{W}$ ), accompagnés de leurs matrices de covariance respectives  $[\mathbf{Q}]$  et  $[\mathbf{R}]$  [14]. Ainsi, les équations d'évolution et de mesure du système deviennent :

$$\mathbf{X}_{k+1} = [\mathbf{A}]\mathbf{X}_k + [\mathbf{B}]\mathbf{U}_k + \mathbf{V}_k \quad (1.31a)$$

$$\mathbf{Y}_k = [\mathbf{C}]\mathbf{X}_k + [\mathbf{D}]\mathbf{U}_k + \mathbf{W}_k \quad (1.31b)$$

Avec :

- $\mathbf{X}_k$  le vecteur d'état à l'instant  $k$  ayant pour covariance la matrice carrée  $[\mathbf{P}]$ ,
- $\mathbf{Y}_k$  le vecteur de mesure à l'instant  $k$ ,
- $\mathbf{U}_k$  le vecteur de commande à l'instant  $k$ ,
- $\mathbf{V}_k$  le vecteur de bruit d'état à l'instant  $k$  ayant pour covariance la matrice carrée  $[\mathbf{Q}]$ ,
- $\mathbf{W}_k$  le vecteur de bruit de mesure à l'instant  $k$  ayant pour covariance la matrice carrée  $[\mathbf{R}]$ .

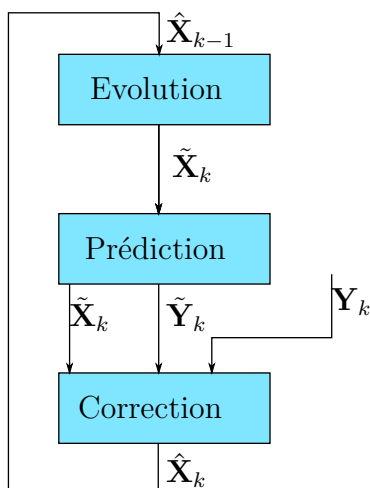


FIGURE 1.9 – Processus récursif du filtre de Kalman

Étape	Filtre de Kalman	Système réel
Évolution	$\tilde{\mathbf{X}}_{k+1} = [\mathbf{A}]\hat{\mathbf{X}}_k + [\mathbf{B}]\mathbf{U}_k$ $[\tilde{\mathbf{P}}_{k+1}] = [\mathbf{A}][\hat{\mathbf{P}}_k][\mathbf{A}]^T + [\mathbf{Q}]$	$\mathbf{X}_{k+1} = [\mathbf{A}]\mathbf{X}_k + [\mathbf{B}]\mathbf{U}_k + \mathbf{V}_k$
Prédiction/mesure	$\tilde{\mathbf{Y}}_k = [\mathbf{C}]\tilde{\mathbf{X}}_k + [\mathbf{D}]\mathbf{U}_k$	$\mathbf{Y}_k = [\mathbf{C}]\mathbf{X}_k + [\mathbf{D}]\mathbf{U}_k + \mathbf{W}_k$
Correction	$\mathbf{E}_k = \mathbf{Y}_k - \tilde{\mathbf{Y}}_k$ $[\mathbf{K}_k] = [\tilde{\mathbf{P}}_k][\mathbf{C}]^T([\mathbf{C}][\tilde{\mathbf{P}}_k][\mathbf{C}]^T + [\mathbf{R}])^{-1}$ $\hat{\mathbf{X}}_k = \tilde{\mathbf{X}}_k + [\mathbf{K}_k]\mathbf{E}_k$ $[\hat{\mathbf{P}}_k] = (\mathbf{I} - [\mathbf{K}_k][\mathbf{C}])[\tilde{\mathbf{P}}_k]$	

TABLE 1.1 – Etapes de fonctionnement du filtre de Kalman

Le filtre de Kalman est un filtre récursif (*i.e.* il utilise la sortie de son estimation corrigée à l'instant  $k$  pour estimer l'état à l'instant  $k + 1$ ) qui peut être représenté sous la forme de la Figure 1.9, avec, pour chaque étape, les opérations décrites en Table 1.1.

Ce filtre vise à minimiser l'erreur quadratique de  $\hat{\mathbf{X}}_k$ .

La plus grosse latitude de réglage du filtre de Kalman, pour un modèle donné, consiste en l'ajustement des coefficients des matrices de covariance de l'estimé  $[\mathbf{P}_0]$  (matrice  $[\mathbf{P}]$  à l'instant initial, permettant une convergence rapide et précise du filtre),  $[\mathbf{Q}]$  et  $[\mathbf{R}]$ .

Pour un système linéaire bien modélisé, ce filtre présente d'excellentes performances. Son plus gros point faible étant cependant son incapacité à traiter les cas non-linéaires. Pour répondre à cela, plusieurs méthodes ont été développées au cours des années. Nous nous concentrerons sur les deux méthodes principales : le filtre de Kalman étendu (*Extended Kalman Filter*, EKF) et l'*Unscented Kalman Filter* (UKF).

## 1.2.5 Observation, fusion de systèmes non-linéaires

Nous allons détailler dans cette partie les deux algorithmes de fusion que nous avons sélectionné pour les systèmes non-linéaires : le filtre EKF et le filtre UKF. Afin d'adresser deux niveaux de non-linéarité différents pour effectuer la comparaison des deux algorithmes, nous nous proposons de prendre pour exemples non-linéaires les deux applications suivantes :

- Nous remplaçons la résistance  $R_{th}$  de notre capteur de rayonnement solaire par une thermistance à coefficient de température négatif (CTN) dont la réponse est

$$R_{th}(T) = R_0 \exp \left[ -B \left( \frac{1}{T_0} - \frac{1}{T} \right) \right] \quad (1.32)$$

avec pour paramètres

$$R_0 = 10000 \pm 5\% \qquad B = 3984 \qquad T_0 = 298$$

Ce système présente une équation de mesure non-linéaire monotone, ainsi la dérivée de cette fonction aura toujours le même signe et sera considérée comme une non-linéarité faible.

- Nous fixons sur une roue dont l'axe est immobile un accéléromètre deux axes  $(x, y)$  selon le schéma en Figure 1.10, la roue tournant à une vitesse  $\dot{\theta}$  et subissant une accélération angulaire  $\ddot{\theta}$ , les deux étant inconnues.

Ce système présente des équations de mesures non-linéaires complexes car faisant appel à des fonctions trigonométriques. Ainsi la dérivée de ces fonctions change de signe en fonction de l'état du système, elles seront donc considérées comme des non-linéarités fortes.

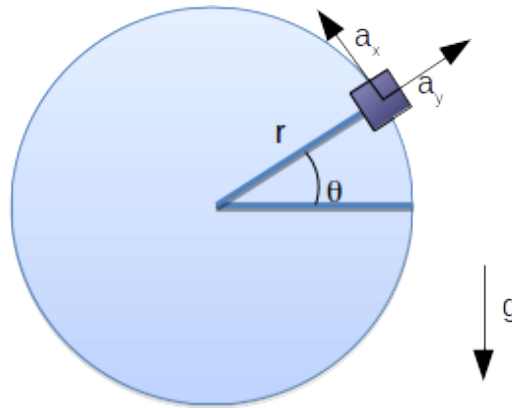


FIGURE 1.10 – Un accéléromètre 2 axes est installé sur une roue pour en déterminer la vitesse

### Le filtre de Kalman étendu (EKF)

Comme évoqué en 1.2.2, la solution la plus fréquente pour traiter un système non linéaire consiste à linéariser ce système autour de son point de fonctionnement. Dans le cas d'un système en représentation d'état, et dont le point de fonctionnement évolue au cours du temps, cela consiste à calculer des matrices Jacobiennes de transition (et de mesure) que nous noterons  $[\mathbf{F}_X]$  et  $[\mathbf{F}_V]$  pour les matrices relatives respectivement à l'évolution de l'état et à l'impact du bruit dans l'équation d'évolution.

De même, pour l'équation de mesure, nous aurons respectivement  $[\mathbf{H}_X]$  et  $[\mathbf{H}_W]$  pour l'état et le bruit.

Ainsi, les équations du système deviennent :

Étape	Filtre de EKF	Système réel
Évolution	$\tilde{\mathbf{X}}_{k+1} = f(\hat{\mathbf{X}}_k, \mathbf{U}_k, 0)$ $[\tilde{\mathbf{P}}_{k+1}] = [\mathbf{F}_{X,k+1}][\hat{\mathbf{P}}_k][\mathbf{F}_{X,k+1}]^T$ $+ [\mathbf{F}_{V,k+1}][\mathbf{Q}][\mathbf{F}_{V,k+1}]^T$	$\mathbf{X}_{k+1} = f(\mathbf{X}_k, \mathbf{U}_k, \mathbf{V}_k)$
Prédiction/mesure	$\tilde{\mathbf{Y}}_k = h(\tilde{\mathbf{X}}_k, \mathbf{U}_k, 0)$	$\mathbf{Y}_k = h(\mathbf{X}_k, \mathbf{U}_k, \mathbf{W}_k)$
Correction	$\mathbf{E}_k = \mathbf{Y}_k - \tilde{\mathbf{Y}}_k$ $[\mathbf{K}_k] = [\tilde{\mathbf{P}}_k][\mathbf{H}_{X,k}]^T([\mathbf{H}_{X,k}][\tilde{\mathbf{P}}_k][\mathbf{H}_{X,k}]^T + [\mathbf{R}])^{-1}$ $\hat{\mathbf{X}}_k = \tilde{\mathbf{X}}_k + [\mathbf{K}_k]\mathbf{E}_k$ $[\hat{\mathbf{P}}_k] = (\mathbf{I} - [\mathbf{K}_k][\mathbf{H}_{X,k}])[\tilde{\mathbf{P}}_k]$	

TABLE 1.2 – Etapes de fonctionnement du filtre de Kalman Etendu

$$\mathbf{X}_{k+1} = f(\mathbf{X}_k, \mathbf{U}_k, \mathbf{V}_k) \quad (1.33a)$$

$$[\mathbf{F}_{X,k+1}] = \frac{\delta f(\mathbf{X}_k, \mathbf{U}_k, \mathbf{V}_k)}{\delta \mathbf{X}_k} \quad (1.33b)$$

$$[\mathbf{F}_{V,k+1}] = \frac{\delta f(\mathbf{X}_k, \mathbf{U}_k, \mathbf{V}_k)}{\delta \mathbf{V}_k} \quad (1.33c)$$

$$\mathbf{Y}_{k+1} = h(\mathbf{X}_k, \mathbf{U}_k, \mathbf{V}_k) \quad (1.33d)$$

$$[\mathbf{H}_{X,k+1}] = \frac{\delta h(\mathbf{X}_k, \mathbf{U}_k, \mathbf{W}_k)}{\delta \mathbf{X}_k} \quad (1.33e)$$

$$[\mathbf{H}_{W,k+1}] = \frac{\delta f(\mathbf{X}_k, \mathbf{U}_k, \mathbf{W}_k)}{\delta \mathbf{W}_k} \quad (1.33f)$$

les fonctions  $f(\cdot)$  et  $h(\cdot)$  étant non-linéaires. Ainsi les matrices jacobienes  $[\mathbf{F}_{X,k}]$  et  $[\mathbf{F}_{V,k}]$  sont utilisées pour la prédiction de  $[\tilde{\mathbf{P}}_{k+1}]$  et la matrice  $[\mathbf{H}_{X,k}]$  est utilisée pour la correction de l'état du système, selon la procédure décrite en Table 1.2.

Cette méthode présente de gros avantages, entre autres, elle est simple à comprendre car s'appuyant directement sur le filtre de Kalman en présentant au filtre un système linéarisé autour d'un point de fonctionnement déterminé par l'état, et évoluant au cours du temps. Elle présente de plus une complexité algorithmique du même ordre que le filtre de Kalman, pour peu que les fonctions  $f(\cdot)$  et  $h(\cdot)$  ainsi que les calculs des jacobienes soient peu gourmands en ressources.

Ce filtre a cependant une faiblesse pour les systèmes présentant de fortes non-linéarités, la propagation de l'erreur et de la densité de probabilité des états (par covariance) s'effectuant de façon linéaire. Ceci peut entraîner des surcompensations et des divergences de l'estimation de l'état dans le pire cas (comme montré en 1.2.5).

Pour parer à cette limitation, l'algorithme *Unscented Kalman Filter* (UKF) [15] a été développé en se basant sur une version mixte entre les filtres à particules [16] et les filtres de Kalman.

### Le filtre *Unscented Kalman Filter* (UKF)

Afin de lever les problèmes liés aux non-linéarités dans la projection des densités de probabilités au travers du modèle par le filtre EKF, le filtre UKF utilise la méthode

*Unscented Transform* pour propager les densités de probabilités directement au travers de la fonction non-linéaire.

***Unscented Transform*** Le principe proposé ici est de remplacer l'approximation linéaire du système et la projection gaussienne au travers de leurs moyennes et covariance par la projection au travers du système non-linéaire d'un ensemble de  $2n + 1$  états pondérés ( $n$  étant la dimension du vecteur d'état) et soigneusement choisis. Il devient alors possible de recalculer les moyennes et covariances des états ou mesure de sortie après avoir subi le comportement non linéaire.

Cette méthode permet ainsi une propagation plus précise de l'erreur d'estimation au travers du système [13][17][14].

Les étapes de fonctionnement de la transformation sont les suivantes :

- Sélectionner les  $2n + 1$  points d'état à propager, représentés par la matrice  $\mathcal{X}^{(i)}$  composée de  $2n + 1$  vecteurs d'état de longueur  $n$  :

$$\mathcal{X}^{(i)} = \begin{cases} \tilde{\mathbf{X}} & , i = 0 \\ \tilde{\mathbf{X}} + \sqrt{(\mu + \lambda)[\mathbf{C}_X]} & , i \in [1, n] \\ \tilde{\mathbf{X}} - \sqrt{(\mu + \lambda)[\mathbf{C}_X]} & , i \in [n + 1, 2n] \end{cases} \quad (1.34)$$

- Transformer les points :

$$\mathcal{Y}^{(i)} = f(\mathcal{X}^{(i)}), i \in [0, 2n] \quad (1.35)$$

- Définir les poids :

$$\omega_i^\mu = \begin{cases} \frac{\lambda}{\lambda+n} & , i = 0 \\ \frac{1}{2(\lambda+n)} & , i \in [1, 2n] \end{cases} \quad (1.36a)$$

$$\omega_i^c = \begin{cases} \omega_0^\mu + 1 - \alpha^2 + \beta & , i = 0 \\ \frac{1}{2(\lambda+n)} & , i \in [1, 2n] \end{cases} \quad (1.36b)$$

- Calculer la moyenne et la covariance du résultat :

$$\tilde{\mathbf{Y}} = \sum_{i=0}^{2n} \omega_i^\mu \mathcal{Y}^{(i)} \quad (1.37a)$$

$$[\tilde{\mathbf{C}}_Y] = \sum_{i=0}^{2n} \omega_i^c (\mathbf{Y}_i - \tilde{\mathbf{Y}})(\mathbf{Y}_i - \tilde{\mathbf{Y}})^T \quad (1.37b)$$

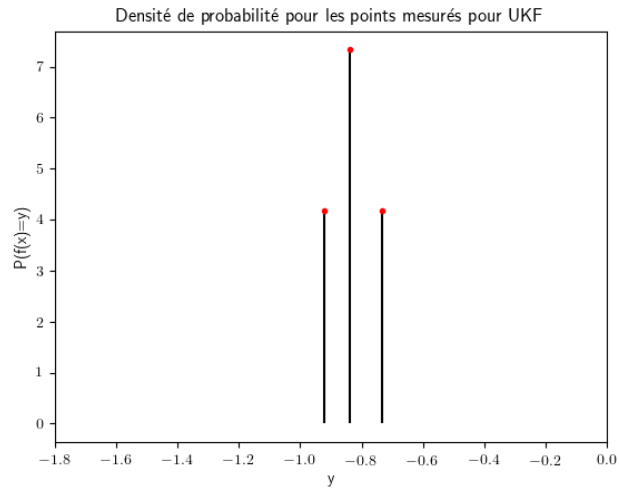
Avec pour paramètres :

- $\alpha$  le facteur d'étalement des points d'estimation par rapport à la moyenne,
- $\beta$  un facteur relatif à la distribution (pour une distribution gaussienne,  $\beta = 2$ ),
- $\lambda = \alpha^2(n + \kappa) - n$ ,
- $\kappa$  le facteur d'échelle,  $\kappa = 3 - n$  dans le cas général.

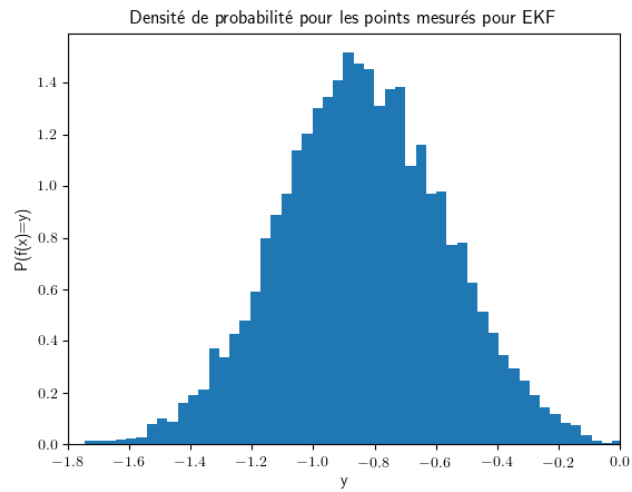
Nous pouvons ainsi appliquer cette méthode pour une fonction non-linéaire à une dimension, par exemple :

$$y = f(x) = \cos(x) \quad (1.38)$$

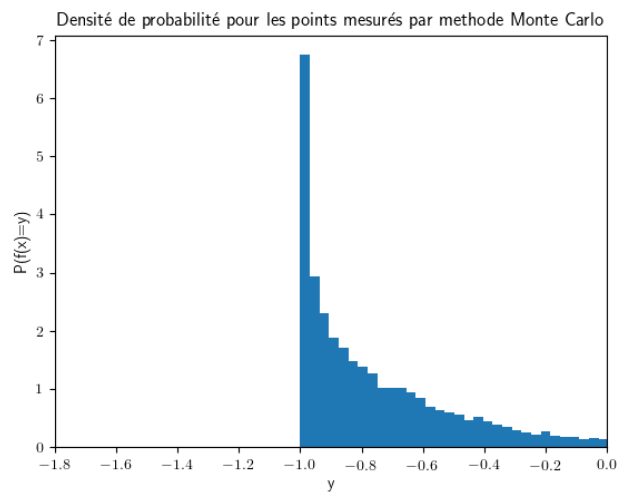
avec en entrée une distribution gaussienne  $x \sim \mathcal{N}(10, 0.5^2)$ . La densité de probabilité obtenue en fonction de la méthode (Monte Carlo, *Unscented Transform* et linéarisation)



(a) Densité de probabilité pour UKF



(b) Densité de probabilité pour EKF



(c) Densité de probabilité pour Monte Carlo

FIGURE 1.11 – Densité de probabilité obtenu sur un cosinus pour  $x \sim \mathcal{N}(10, 0.5^2)$

Méthode	Moyenne	Covariance
Monte Carlo	-0.7354	0.07860
<i>Unscented Transform</i>	-0.734450	0.0960
Linéarisation	-0.8390	0.07399

TABLE 1.3 – Données statistiques propagées dans  $f(x) = \cos(x)$  pour  $x \sim \mathcal{N}(10, 0.5^2)$

est présentée en Figure 1.11, ainsi que les moyennes et covariances obtenues après les différentes méthodes (Table 1.3).

Nous remarquons ainsi que la méthode *Unscented Transform* fournit une approximation plus précise de la linéarisation locale, tout en ayant une charge calculatoire nettement plus faible que la méthode Monte Carlo.

A partir de là, la méthodologie pour obtenir le filtre de Kalman UKF est la suivante :

— Évolution :

— Générer un jeu de points pondérés pour l'état

$$\mathcal{X}_k = \left[ \hat{\mathbf{X}}_k, \hat{\mathbf{X}}_k + \sqrt{(n + \lambda)[\hat{\mathbf{P}}_k]}, \hat{\mathbf{X}}_k - \sqrt{(n + \lambda)[\hat{\mathbf{P}}_k]} \right] \quad (1.39)$$

Avec  $\mathcal{X}_k$  une matrice de dimension  $(n, 2n + 1)$  et représentant les  $2n + 1$  états à propager (pondérés par  $\omega^c$  et  $\omega^\mu$  calculés précédemment et fixes pour un système donné).

— Propager l'état au travers de sa fonction d'évolution

$$\tilde{\mathcal{X}}_{k+1}^{(i)} = f(\mathcal{X}_k^{(i)}, \mathbf{U}_k, 0), i \in [0, 2n] \quad (1.40)$$

— Puis calculer ses statistiques avec les méthodes de l'*Unscented Transform*

$$\tilde{\mathbf{X}}_{k+1} = \sum_{i=0}^{2n} \omega_i^\mu \tilde{\mathcal{X}}_{k+1}^{(i)} \quad (1.41a)$$

$$[\tilde{\mathbf{P}}_{k+1}] = \sum_{i=0}^{2n} \omega_i^c (\tilde{\mathcal{X}}_{k+1}^{(i)} - \tilde{\mathbf{X}}_{k+1})(\tilde{\mathcal{X}}_{k+1}^{(i)} - \tilde{\mathbf{X}}_{k+1})^T + [\mathbf{Q}] \quad (1.41b)$$

$$(1.41c)$$

— Projection de la mesure :

— Générer un jeu de points pour l'état estimé

$$\tilde{\mathcal{X}}_{k+1} = \left[ \tilde{\mathbf{X}}_{k+1}, \tilde{\mathbf{X}}_{k+1} + \sqrt{(n + \lambda)[\tilde{\mathbf{P}}_{k+1}]}, \tilde{\mathbf{X}}_{k+1} - \sqrt{(n + \lambda)[\tilde{\mathbf{P}}_{k+1}]} \right] \quad (1.42)$$

— Propager les points dans la fonction de mesure

$$\tilde{\mathcal{Y}}_{k+1}^{(i)} = h(\tilde{\mathcal{X}}_{k+1}^{(i)}, \mathbf{U}_{k+1}, 0), i \in [0, 2n] \quad (1.43)$$

— Estimer l'espérance et la covariance de la mesure

$$\tilde{\mathbf{Y}}_{k+1} = \sum_{i=0}^{2n} \omega_i^\mu \tilde{\mathcal{Y}}_{k+1}^{(i)} \quad (1.44a)$$

$$[\tilde{\mathbf{P}}_{yy,k+1}] = \sum_{i=0}^{2n} \omega_i^c (\tilde{\mathcal{Y}}_{k+1}^{(i)} - \tilde{\mathbf{Y}}_{k+1})(\tilde{\mathcal{Y}}_{k+1}^{(i)} - \tilde{\mathbf{Y}}_{k+1})^T + [\mathbf{R}] \quad (1.44b)$$



— ainsi que la covariance croisée état/mesure

$$[\tilde{\mathbf{P}}_{xy,k+1}] = \sum_{i=0}^{2n} \omega_i^c (\tilde{\mathcal{X}}_{k+1}^{(i)} - \tilde{\mathbf{X}}_{k+1})(\tilde{\mathcal{Y}}_{k+1}^{(i)} - \tilde{\mathbf{Y}}_{k+1})^T \quad (1.45)$$

— On en déduit finalement le gain de Kalman

$$[\mathbf{K}_{k+1}] = [\tilde{\mathbf{P}}_{xy,k+1}][\tilde{\mathbf{P}}_{yy,k+1}]^{-1} \quad (1.46)$$

— Pour finalement corriger l'état et sa covariance à partir de la mesure  $\mathbf{Y}_{k+1}$

$$\hat{\mathbf{X}}_{k+1} = \tilde{\mathbf{X}}_{k+1} + [\mathbf{K}_{k+1}](\mathbf{Y}_{k+1} - \tilde{\mathbf{Y}}_{k+1}) \quad (1.47a)$$

$$[\hat{\mathbf{P}}_{k+1}] = [\tilde{\mathbf{P}}_{k+1}] - [\mathbf{K}_{k+1}][\tilde{\mathbf{P}}_{yy,k+1}][\mathbf{K}_{k+1}]^T \quad (1.47b)$$

Bien que présentant une charge de calculs plus importante, le filtre UKF reste plus proche de la densité de probabilité réelle du système, ce qui lui permet de converger plus rapidement et d'obtenir une estimation plus précise que les deux méthodes précédentes en cas de forte non-linéarité, comme nous allons le montrer au cours de la prochaine section.

### Filtres de Kalman hybrides (HEKF/HUKF)

Les différents filtres de Kalman proposés jusqu'à présent sont soit adaptés à des systèmes linéaires et peu demandeurs en ressources, soit à des systèmes non-linéaires et nécessitant des ressources importantes. Cependant, nous avons vu que la représentation d'état décrit un système selon deux équations :

- une équation d'évolution,
- une équation de mesure.

Les filtres proposés supposent que ces deux équations soient linéaires ou non-linéaires. Ainsi, le cas d'un système hybride, dont l'une des équations est linéaire et l'autre non-linéaire, nous pousse à utiliser des solutions de filtre non-linéaire pour les deux équations, générant ainsi un surcoût calculatoire.

La solution que nous proposons consiste à dériver des deux catégories de filtres précédentes une nouvelle catégorie de filtres dits hybrides, utilisant la résolution linéaire d'un filtre KF pour l'équation linéaire et la résolution EKF ou UKF pour l'équation non-linéaire. Nous obtenons ainsi les filtres *Hybrid Extended Kalman Filter* (HEKF) et *Hybrid Unscented Kalman Filter* (HUKF). Cette solution permet de garantir le meilleur compromis précision/performances pour l'estimation de l'état d'un tel système.

Nous allons maintenant comparer le comportement des filtres proposés pour les deux systèmes présentés précédemment :

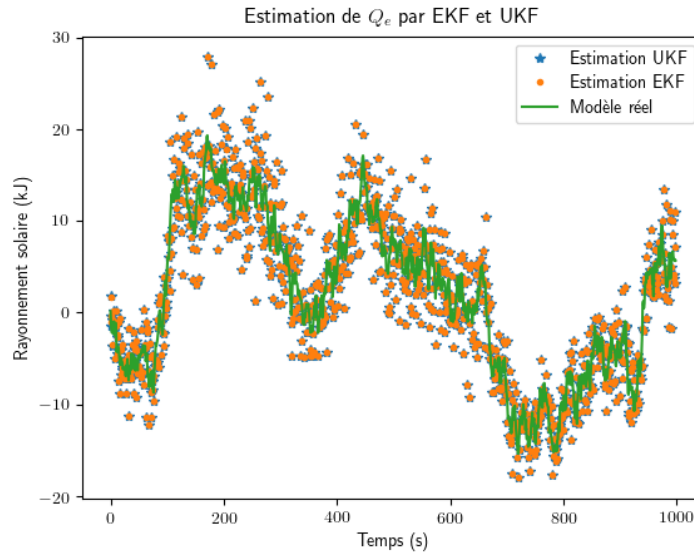
- **Le capteur de rayonnement solaire à CTN** pour une non-linéarité faible,
- **Le capteur accéléromètre sur roue** pour une non-linéarité forte.

### Capteur de rayonnement solaire à CTN

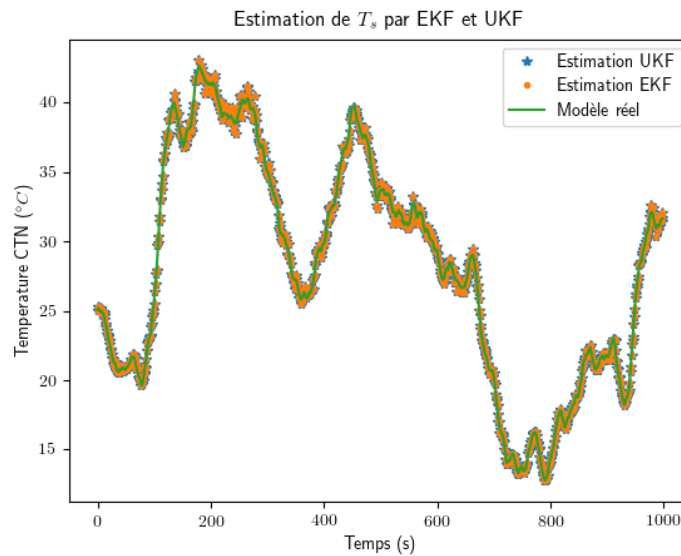
Ce système permet d'illustrer le cas où seule l'équation de sortie est non-linéaire, le comportement dynamique du système restant linéaire. Ainsi, pour le filtre EKF, nous devons calculer la matrice jacobienne de mesure  $[\mathbf{H}]$  qui aura pour valeur (en fonction de la température du système  $T_s$  en  $^{\circ}\text{C}$ ) :

$$[\mathbf{H}](T_s) = \left[ 0, \frac{-B}{(T_s + 273)^2} \exp \left( -B \left( \frac{1}{T_0} - \frac{1}{T_s + 273} \right) \right) \right] \quad (1.48)$$

Lorsque nous initialisons les filtres EKF et UKF dans le même état, nous obtenons pour une évolution bruitée de  $Q_e$  et une valeur fixe de  $T_{amb}$  l'estimation de  $Q_e$  (Figure 1.12a) et de  $T_s$  (Figure 1.12b).



(a) Estimation de  $Q_e$



(b) Estimation de  $T_s$

FIGURE 1.12 – Résultats d'estimation EKF et UKF sur l'estimateur de rayonnement non-linéaire

Nous remarquons ainsi que pour cette application, le filtre EKF et le filtre UKF ont des performances équivalentes et fournissent une estimation cohérente avec la réalité terrain. Dans ce cas, le filtre UKF ayant généralement une complexité plus importante, il est donc préférable d'utiliser un filtre EKF.

## Capteur accéléromètre sur roue

Pour cet exemple, l'évolution du système est linéaire et son équation de sortie est fortement non-linéaire (fonctions trigonométriques).

L'état du système est représenté par le vecteur  $\mathbf{X}$  :

$$\mathbf{X} = \begin{pmatrix} \theta \\ \dot{\theta} \\ \ddot{\theta} \end{pmatrix} \quad (1.49)$$

avec

- $\theta$  l'angle de rotation de la roue (en *rad*),
- $\dot{\theta}$  la vitesse angulaire de la roue (en *rad/s*),
- $\ddot{\theta}$  l'accélération angulaire de la roue (en *rad/s<sup>2</sup>*).

La matrice d'évolution  $[\mathbf{A}]$  est :

$$[\mathbf{A}] = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \quad (1.50)$$

Le vecteur de sortie est représenté par

$$\mathbf{Y} = \begin{pmatrix} a_x \\ a_y \end{pmatrix} = \begin{pmatrix} g \cos(\theta) + r\ddot{\theta} \\ -g \sin(\theta) + \dot{\theta}^2 \end{pmatrix} \quad (1.51)$$

avec  $g = 9.80N$ , l'accélération gravitationnelle et  $r = 0.5m$  le rayon de la roue.

L'accéléromètre sur l'axe  $x$  (tangenciel à la roue) mesure l'accélération gravitationnelle et l'accélération tangentielle alors que l'accéléromètre sur l'axe  $y$  (dans l'axe de la roue) mesure l'accélération gravitationnelle et la force centripète exercée par la roue en rotation.

Pour le filtre EKF, nous utilisons la matrice jacobienne de mesure :

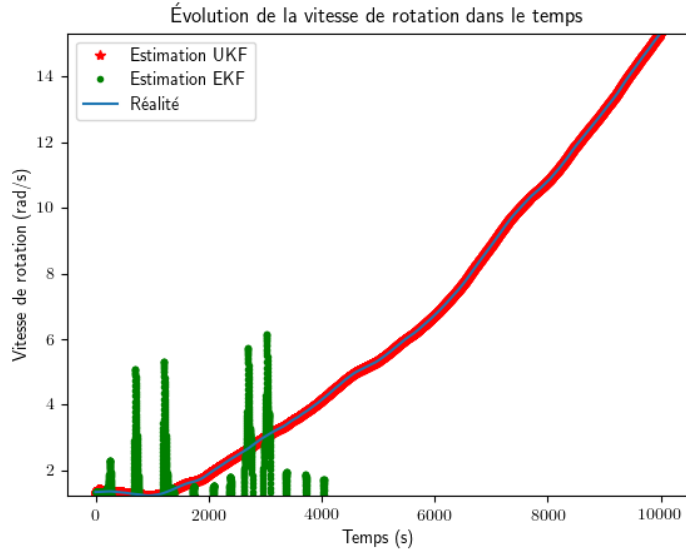
$$[\mathbf{H}](\theta) = \begin{bmatrix} \frac{\delta a_x}{\delta \theta} & \frac{\delta a_x}{\delta \dot{\theta}} & \frac{\delta a_x}{\delta \ddot{\theta}} \\ \frac{\delta a_y}{\delta \theta} & \frac{\delta a_y}{\delta \dot{\theta}} & \frac{\delta a_y}{\delta \ddot{\theta}} \end{bmatrix} = \begin{bmatrix} g \sin(\theta) & 0 & r \\ -g \cos(\theta) & 2 & 0 \end{bmatrix} \quad (1.52)$$

Nous obtenons pour estimation de la vitesse de rotation la Figure 1.13a et pour l'estimation de l'accélération angulaire la Figure 1.13b, le filtre EKF ayant été initialisé avec l'état initial du modèle pour limiter la divergence à l'initialisation.

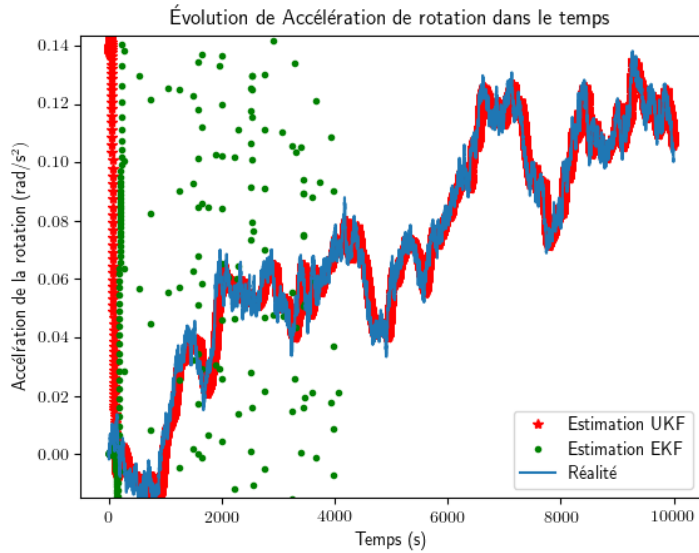
Nous remarquons ainsi que le filtre EKF diverge rapidement sur cette application (en moins de 10s), jusqu'à atteindre une singularité numérique aux alentours de 4000s empêchant la poursuite des calculs, alors que le filtre UFK arrive à maintenir l'estimation lorsqu'il a convergé vers l'état du système.

### 1.2.6 Stratégie de calibration et représentation

Afin de permettre à un système de mettre à jour de façon automatique sa calibration, nous avons choisi d'appliquer une stratégie se basant sur l'utilisation de deux filtres de Kalman complémentaires : un filtre chargé d'estimer l'état interne du système et prenant comme entrée de commande les paramètres de calibration, un second filtre chargé d'estimer les paramètres de calibration n'étant activé que lorsque les conditions de fonctionnement du système sont connues. Ainsi, si nous notons  $\mathbf{\Gamma}$  l'ensemble des paramètres à



(a) Estimation de la vitesse angulaire



(b) Estimation de l'accélération angulaire

FIGURE 1.13 – Comparaison des filtres EKF et UKF pour l'estimation de l'état de la roue

estimer et  $\Phi$  l'ensemble des paramètres de calibration du modèle, nous obtenons l'architecture présentée en Figure 1.14. Lors de la phase de calibration (phase 1) les paramètres  $\Phi$  sont fixés et connus, l'estimateur est alors chargé de déterminer les paramètres de calibration  $\Gamma$ . Dans un second temps, lorsque le système est calibré, il fonctionne en mode estimation (phase 2), et estime les valeurs de  $\Phi$  à partir des paramètres de calibration fixes déterminés précédemment.

Cette méthodologie se base sur le besoin de travailler avec des systèmes observables pour garantir des estimations non biaisées, ainsi les deux sous-systèmes sont suffisamment définis par leurs conditions propres pour être observables.

De plus, comme nous le verrons dans la section 2.5, cette solution permet de diminuer la taille du vecteur d'état, ce qui diminue de façon importante la complexité des calculs à

effectuer et les rend donc embarquables sur une architecture à faible puissance de calculs tel que les micro-contrôleurs.

Selon les caractéristiques des paramètres du système, ces filtres pourront être linéaires, non-linéaires ou hybrides.

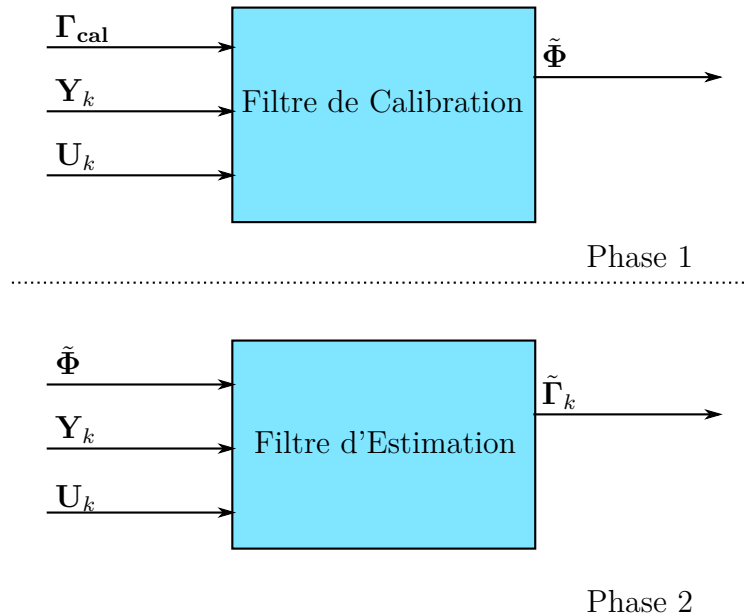


FIGURE 1.14 – Architecture de gestion de la calibration

Dans le cas de notre capteur de rayonnement solaire, nous pouvons, par exemple, fixer  $\Phi = (R_0)$  et  $\Gamma = (Q_e)$ .

Lors de la calibration du capteur, nous positionnons celui-ci dans le noir. Nous sommes donc en conditions contrôlées, avec  $Q_e = 0$ . Le premier estimateur peut alors déterminer la valeur du paramètre  $R_0$ .

Une fois cette étape effectuée, la valeur  $R_0$  est connue et figée, le second estimateur peut alors déterminer la valeur de  $Q_e$  quelles que soient les conditions de fonctionnement du système.

### 1.3 Conclusion

Au cours de ce chapitre, nous avons exploré la problématique de la fusion de données multi-physiques pour les capteurs.

Dans un premier temps, nous nous sommes concentrés sur la modélisation, avec les problématiques liées à la linéarité, à la dispersion et au vieillissement des capteurs, puis, nous nous sommes intéressés aux modes de représentations qui étaient à notre disposition, pour finalement se concentrer sur la représentation d'état qui est la plus appropriée dans le cas de systèmes multi-entrées, multi-sorties et à états internes cachés.

Finalement, nous nous sommes intéressés aux observateurs nous permettant d'estimer l'état interne de notre modèle, et en particulier sur les filtres de Kalman.

En lien avec la problématique de non-linéarité de la plupart des systèmes physiques, nous avons développé deux classes de filtres de Kalman non-linéaires : l'EKF et l'UKF.

Bien que le premier présente un avantage de simplicité de mise en place et de compréhension, nous avons montré que le filtre UKF présentait un net avantage en termes de précision face à son concurrent.

Nous avons ainsi présenté un système de filtres hybrides permettant d'obtenir de meilleures performances pour les systèmes présentant une équation linéaire et l'autre non-linéaire.

L'évolution de la complexité des systèmes et de la capacité de calculs nous permet désormais de réaliser des capteurs moins coûteux en effectuant, de façon algorithmique, la fusion de données d'un ensemble de sous-capteurs moins sélectifs. Ainsi, un capteur de qualité des fluides peut maintenant effectuer des mesures aussi variées que la température, la puissance d'émission et de transmission optique de LEDs, la conductivité ou la capacité du fluide [18].

### 1.3.1 Notre fil d'Ariane : le capteur UQS

Les solutions que nous allons proposer s'appuient sur une identification des paramètres d'influence des différents éléments constitutifs du capteur, une modélisation des influences et une fusion récursive des données permettant d'estimer la grandeur d'influence représentant la mesure que nous souhaitons exploiter. Pour cela, nous allons d'abord détailler la modélisation des éléments constitutifs, puis la représentation du système. Cela nous permettra par la suite de déterminer si la grandeur que nous souhaitons mesurer est observable. Finalement nous traiterons de l'estimation et du filtrage de l'information.

L'un des exemples de cette nouvelle méthodologie est le capteur UQS (*Urea Quality Sensor*) que nous détaillerons au chapitre 3, développé par *MEAS-France / TE Connectivity*, dont le but est d'effectuer une mesure embarquée en temps-réel de la concentration d'urée en solution aqueuse dans le réservoir dédié à cet effet sur les poids lourds et engins de chantier. Pour s'acquitter de cette tâche, ce capteur est composé de LEDs (*Light Emitting Diodes*) infrarouges à plusieurs longueurs d'ondes, de photodétecteurs et de capteurs de température de type CTN (résistances à Coefficient de Température Négatif) (Figure 1.15).

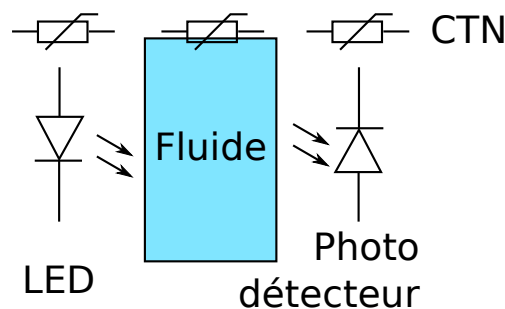


FIGURE 1.15 – Principe de base du capteur UQS

Bien que performantes, les techniques que nous avons abordé au cours de ce chapitre présentent cependant une contrepartie en termes de charge de calcul et d'occupation mémoire. Cela peut s'avérer sans grand désagrément lorsque le programme est exécuté sur une machine moderne. L'objectif de réaliser des capteurs embarqués nous contraint cependant à limiter nos ressources, pour des raisons de coût ou de consommation. Ainsi, dans le prochain chapitre, nous allons nous intéresser à la réduction de la charge de notre

algorithme de fusion sur le processeur sur lequel il s'exécute afin de pouvoir l'embarquer au sein d'un capteur intelligent.

# Bibliographie

- [1] R. MOOS. A brief overview on automotive exhaust gas sensors based on electroceramics. *Int. J. Appl. Ceram. Technol*, pages 401–413, 2005.
- [2] Jing GAO. *Etude et Mise au Point d'un Capteur de Gaz Pour la Détection Sélective de NOx en Pot d'Echappement Automobile*. PhD thesis, 2011.
- [3] P.VERNOUX A.HADJAR, J.P VIRICELLE. Complementary study and optimization of potentiometric nox sensor and nh3 detection, July 2012.
- [4] M.Schweizer-Berberich et al. Pulsed mode of operation and artificial neural network evaluation for improving the co selectivity of sno2 gas sensors. *Sensors and Actuators B : Chemical*, 65 :91–93, June 2000.
- [5] J. M. Kahn ; R. H. Katz ; K. S. J. Pister. Next century challenges : Mobile networking for “smart dust”. *Proceeding MobiCom '99 Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, pages 271–278, 1999.
- [6] The Economist. Where the smart is, June 2016.
- [7] Kalman R. E. A new approach to linear filtering and prediction problems. *Transaction of the ASME- Journal of Basic Engineering*, 82 (series D) :35–45, 1960.
- [8] B.F. Spencer Jr. ; Manuel E. Ruiz-Sandoval ; Narito Kurata. Smart sensing technology : Opportunities and challenges. *Structural Control Health Monitoring journal*, 11 :349–368, 2004.
- [9] Yves Granjon. *Automatique, Systèmes linéaires, non linéaires, à temps continu, à temps discret, représentation d'états*. Dunod, 2nde edition, 2010.
- [10] B.M. Brown. *The Mathematical Therory of Linear Systems*. Chapman & Hall LTD., 1961.
- [11] H. Wang ; S. Dong. Adaptive fusion design using multiscale unscented kalman filter approach for multisensor data fusion. *Mathermatical Problems in Engineering*, 2015 :10, 2015.
- [12] Y. Huang et al. *Journal of Bionic Engineering*, Volume 4, Issue 1 :53–62, March 2007.
- [13] Fredrik Orderud. Comparison of kalman filter estimation approaches for state space models with nonlinear measurements. 2005.
- [14] Robert Grover Brown ; Patrick Y.C. Hwang. *Introduction to Random Signal and Applied Kalman Filtering*. Wiley, 4th edition, 2012.
- [15] S.J. Julier ; J.K. Uhlmann. A new extension of the kalman filter to nonlinear systems. *Signal Processing, Sensor Fusion, and Target Recognition VI*, 3068, 1997.
- [16] P. Del Moral ; A. Doucet. Particle methods : An introduction with applications. *ESAIM : PROCEEDINGS*, 44 :1–46, January 2014.



- [17] Eric A. Wan and Rudolph Van Der Merwe. The unscented kalman filter for nonlinear estimation. pages 153–158, 2000.
- [18] Hao Wu ; Chao Tan ; Feng Dong. An on-line adaptive estimation method for water holdup measurement in oil–water two-phase flow with a conductance/capacitance sensor. *Measurement Science and Technology*, 27(7), July 2016.

# Chapitre 2

## Vers une sobriété technique, ou comment simplifier pour embarquer

Au cours du chapitre précédent, nous avons vu que, sous hypothèse de ressources suffisantes, il était possible de réaliser un capteur multi-physique intelligent basant sa mesure sur la fusion de données provenant de plusieurs de ses éléments constitutifs. Cette solution présente, entre autres, l'avantage de pouvoir embarquer un capteur sur un équipement dont les conditions de fonctionnement ne sont pas contrôlées de façon précise.

Les algorithmes permettant de résoudre ces problématiques sont cependant parfois compliqués à exécuter en temps réel sur une machine moderne (par exemple, le filtre particulière consomme une grande quantité de ressources pour obtenir une précision et une vitesse de convergence satisfaisantes). À partir de ce constat, la question d'embarquer des algorithmes complexes sur capteurs se pose. En effet, les ressources disponibles sur le capteur sont nettement inférieures aux ressources d'un ordinateur moderne, pour des raisons de coût, d'encombrement et de consommation électrique. Ainsi, comment est-il possible de sélectionner et d'adapter une solution présentée au chapitre 1.2 pour qu'elle fonctionne sur un capteur ?

Au cours de ce chapitre, nous allons nous intéresser dans un premier temps aux questions de limitation des ressources liées aux applications embarquées, à la complexité des algorithmes utilisés et de leurs implications en termes de temps de calcul. Dans une seconde section, nous nous attarderons sur les notions de temps-réel et de criticité avant de présenter les solutions proposées afin de diminuer la charge de calculs, et ainsi permettre à notre capteur d'embarquer de l'intelligence.

### 2.1 Un capteur au sein d'un système : la relation au temps

Plus qu'un système embarqué multi-physique, un capteur est lui-même un élément constitutif d'un système. Son rôle est de renseigner ce système sur l'état et l'évolution de son environnement. En ce sens, il se place en tant que passerelle entre les deux et subit par la même occasion une double contrainte : il doit être en mesure de transmettre les informations à la vitesse demandée par le système, tout en prenant en compte la dynamique de son environnement le plus fidèlement possible. On parle alors de capteur temps-réel.

### 2.1.1 Notions de temps réel

La notion de temps réel peut être définie comme :

En informatique, on parle d'un système temps réel lorsque ce système est capable de contrôler (ou piloter) un procédé physique à une vitesse adaptée à l'évolution du procédé contrôlé[1].

Pour une interface homme-machine par exemple, une réponse inférieure à 100ms sera perçue comme temps-réel[2], alors que pour la mesure de température d'un bâtiment, un temps de réponse de 10s sera lui aussi temps-réel.

Afin d'être considéré comme un système temps-réel, notre capteur doit donc être en mesure d'effectuer ses mesures et de fusionner les données, le tout en un temps inférieur à la période de mesure demandée (imposée par le système ou l'environnement, le temps le plus court étant celui qui fixe la contrainte). Le temps de calcul et donc la complexité des algorithmes, ou bien la puissance du micro-contrôleur, doit être adapté afin de satisfaire cette contrainte, quelles que soient les conditions.

### 2.1.2 Risques

En cas de non respect de la contrainte de temps-réel, le capteur peut se trouver dans le cas où sa fréquence d'échantillonnage, et donc le modèle d'évolution du système, est instable[3]. Il peut aussi perdre des échantillons de la mesure si le processus d'échantillonnage est indépendant des calculs. Dans les deux cas, un problème de consistance des données entre le modèle et la réalité est probable, on parle alors de désynchronisation.

De plus, un calcul trop long et une désynchronisation peuvent entraîner une perte de données en sortie ou la transmission de données erronées. Les risques de tels événements et leur criticité doivent être évalués pour déterminer une marge de sécurité vis-à-vis du temps de calcul. Ainsi, pour une forte criticité, il faudra garantir un temps réel du capteur quelles que soient les conditions, on parle alors de temps réel "dur". Dans le cas d'une plus faible criticité, on pourra se permettre des désynchronisations occasionnelles, on parle alors de temps réel "mou".

Par exemple, un capteur de vitesse de roue sur une voiture a une forte criticité si on le lie au calculateur d'ABS, ou de suspension [4], avec un temps de réponse devant être le plus faible possible (disons 10ms). Si on le lie au tableau de bord pour afficher la vitesse du véhicule, la criticité diminue ainsi et la contrainte temps réel peut être relâchée à un temps de réponse de 100ms sans détérioration des performances.

### 2.1.3 Verrous

Lorsque la criticité d'un système nous impose un comportement temps-réel avec des temps de réponse faibles, nous pouvons faire face à plusieurs verrous qu'il nous faudra lever avant d'atteindre notre objectif. Parmi ceux-là, nous pouvons trouver :

- Une fréquence d'échantillonnage mal dimensionnée vis-à-vis de l'évolution de l'environnement : un sous-échantillonnage nous fait perdre des informations importantes tandis qu'un sur-échantillonnage augmente le temps de calcul de façon inutile,
- Un modèle ou des algorithmes trop complexes pour être exécutés dans le temps imparti sur le micro-contrôleur sélectionné,

- Une mauvaise sélection de micro-contrôleur, trop petit en termes de mémoire, ou trop faible en vitesse de calcul pour exécuter les calculs minimaux requis dans le temps imparti.

Ainsi dans le premier cas, une étude du comportement du système peut permettre de trouver le meilleur compromis en termes de période d'échantillonnage pour l'environnement déterminé. Dans le second cas, il existe des techniques de simplification que nous détaillerons en section 2.5. Finalement, si l'optimisation d'algorithmes ne suffit pas, le remplacement de l'unité de calcul par une version plus adaptée est incontournable (à moins de réévaluer les exigences de temps de réponse à la baisse).

## 2.2 Quid des limitations techniques

Les limitations techniques liées aux systèmes embarqués (et par extension aux capteurs intelligents) proviennent en grande partie des choix architecturaux inhérents à ces systèmes. Ainsi, un système embarqué se distingue principalement par son alimentation électrique autonome (généralement sur batteries), ce qui implique une recherche de la diminution de la consommation électrique. En outre, il fait souvent parti d'un système plus grand et donc doit tendre vers une diminution des coûts de production.

### 2.2.1 Limitations liées aux systèmes embarqués

Ainsi les systèmes embarqués sont principalement caractérisés par des ressources matérielles limitées. Un capteur intelligent à très faible coût sera donc probablement construit autour d'un micro-contrôleur simple coeur, avec une architecture 8 bits et une fréquence de calcul inférieure à 20MHz. La Figure 2.1 montre que le coût aussi bien financier qu'énergétique rend possible l'utilisations de puces de plus en plus complexes, ainsi les processeurs 32 bits sont aussi utilisés que ceux de 8 bits de nos jours.

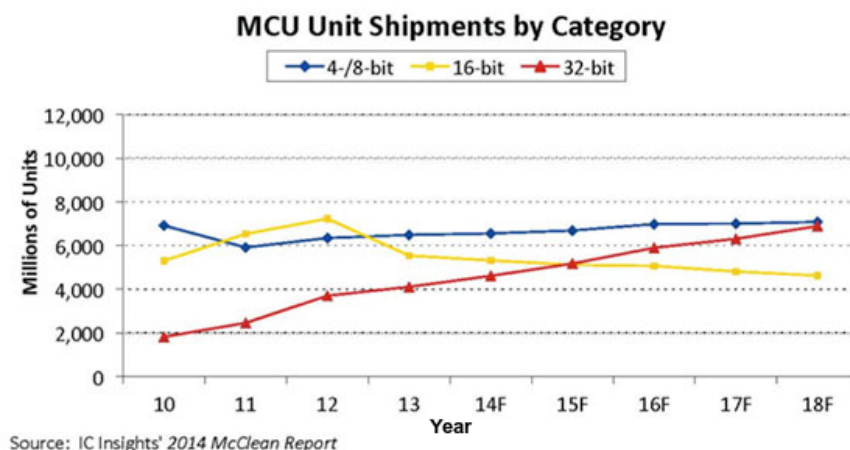


FIGURE 2.1 – Évolution des parts de marché des micro-contrôleurs en fonction de leur architecture [5]

De plus, les périphériques permettant d'accélérer les calculs tels que les unités à virgule flottante (*Floating Point Unit*, FPU [6]) souvent absentes dans le passé car coûteuses et gourmandes en énergie, deviennent de plus en plus accessibles [7]. Afin d'estimer la

Processeur	Flops simple précision	Ops virgule fixe 32 bits
9S08 8 bits / 16MHz	$\approx 17000$	$\approx 3M$
STMF103RB 32 bits / 72MHz	$\approx 340000$	$\approx 32M$
STMF4x 32 bits / 216 MHz / FPU	$\approx 1M$ sans FPU, $\approx 12M$	$\approx 100M$

TABLE 2.1 – Performances

faisabilité d’algorithmes sur les capteurs embarqués, nous avons évalué la capacité de quelques architectures représentatives du marché et des possibilités offertes :

- le M9S08 comme représentant de petits processeurs 8 bits
- les STMF1xx comme représentant de processeurs 32 bits sans unité de virgule flottante
- les STMF4xx comme représentants de processeurs 32 bits avec unités de calcul en virgule flottante

Un simple banc de test a été réalisé sur un M9S08DZ128 et un STMF103RB permettant de mesurer le temps d’exécution en nombre de cycles d’horloges de plusieurs multiplications/accumulations avec différentes résolutions de calcul (virgule fixe, et flottant simple précision). Nous utilisons les résultats du benchmark [8] pour extrapoler les performances dans le cas de l’utilisation de l’unité flottante du STMF4xx.

La Table 2.1 donne les ordres de grandeur du nombre d’opérations (arithmétiques simples : multiplications par constantes, multiplications, additions) par seconde, mesurés sur les différentes architectures.

On remarque que l’effort de codage des calculs en virgule fixe permet des gains énormes en puissance de calcul. Cependant les algorithmes évoqués par la suite peuvent perdre leur propriétés par cette dégradation de la résolution, spécialement les filtres de Kalman utilisant des inversions de matrices où le passage en virgule fixe peut rendre l’algorithme instable. Nous évaluerons donc par la suite uniquement le pire cas de l’utilisation de calculs en virgule flottante simple précision.

Une limitation importante est celle de la taille de la mémoire embarquée (quelques kilo-octets de programme et de mémoire vive) qui contraint les calculs les plus exigeants à être déportés. Bien que les processeurs soient déclinés sous différentes tailles de mémoire, sans impact sur l’architecture du code et du système (souvent seul le coût est modifié, dans un même boîtier, pour un même système), celle-ci peut limiter l’utilisation de certains algorithmes. L’empreinte mémoire des algorithmes n’est pas étudiée dans ce mémoire car une première évaluation a montré que ce critère n’était pas critique dans le cas des applications envisagées sur une puce comportant 128 ko de Flash et 6ko de RAM.

En revanche, la durée d’exécution des algorithmes constitue une contrainte très critique, les sections suivantes évaluent la complexité des algorithmes d’estimation envisagés afin de donner un outil pour leur dimensionnement.

## 2.3 Complexité algorithmique

L’étude de la vitesse d’exécution ou de l’occupation mémoire d’un algorithme est une étape importante pour comparer les algorithmes entre eux et sélectionner celui qui conviendra le mieux pour une application donnée.

Selon le type d’opération effectuée, ou l’architecture matérielle utilisée, nous pouvons dresser plusieurs types d’indicateurs de complexité. On trouvera ainsi pour les plus

connus :

- Le nombre de lignes de code exécutées,
- Le nombre d'opérations à virgule flottante exécutées (*Floating Point Operations*, FLOP),
- Nombre d'accès mémoire...

D'après [9], l'analyse asymptotique mathématique d'un algorithme permet de donner un indicateur plus fiable qu'une mesure de performances effectuée sur une architecture donnée avec un jeu de données fixé.

Ainsi, l'opération consiste à prendre un indicateur numérique donné, en fonction d'une grandeur d'entrée, notée  $n$  (par exemple le nombre de points dans un vecteur) et de définir la complexité de l'algorithme  $T(f(n))$ .

Par exemple, pour le calcul de la moyenne d'un vecteur de longueur  $n$ , avec l'algorithme 1.

<p><b>Algorithme 1</b> : Calcul de la moyenne d'un vecteur</p> <hr/> <p><b>Fonction</b> <i>Moyenne (vecteur)</i></p> <p style="padding-left: 20px;"><b>Entrée</b> : vecteur Vecteur dont on veut calculer la moyenne</p> <p style="padding-left: 20px;"><b>Sortie</b> : moyenne Moyenne du vecteur d'entrée</p> <p style="padding-left: 20px;">somme=0;</p> <p style="padding-left: 20px;"><b>Pour</b> <math>i</math> <b>de</b> 1 <b>à</b> longueur(<i>vecteur</i>) <b>faire</b></p> <p style="padding-left: 40px;">  somme = somme + vecteur[i];</p> <p style="padding-left: 20px;"><b>FinPour</b></p> <p style="padding-left: 20px;">moyenne = <math>\frac{\text{somme}}{\text{longueur}(\text{vecteur})}</math>;</p> <p><b>FinFonction</b></p>
---

Nous obtenons, ainsi :

- Pour le nombre de lignes exécutées :  $T(3 + 2n)$ ,
- Pour le nombre d'opérations à virgule flottante :  $T(2 + 2n)$ .

Nous pouvons aussi utiliser l'indicateur  $\mathcal{O}(n)$  comme indicateur de complexité asymptotique de l'algorithme. Ainsi, dans les deux cas :

$$\lim_{n \rightarrow +\infty} T(f(n)) = \mathcal{O}(2n) \tag{2.1}$$

à interpréter comme le fait que la complexité augmente de façon proportionnelle à  $2n$ .

Nous pouvons ainsi appliquer cette démarche dans le cas qui nous intéresse pour étudier la complexité algorithmique des différentes opérations de base pour notre fusion de données afin de nous aider à dresser une carte des complexités de nos différents filtres de Kalman.

Les opérations mathématiques de base liées à notre filtre de Kalman sont :

- La multiplication de matrices,
- L'addition de vecteurs,
- L'addition de matrices,
- La transposition de matrice,
- L'inversion de matrice.

De plus, le filtre UKF utilise les opérations spécifiques suivantes :

- Le calcul d'un vecteur moyen depuis un jeu de vecteurs,
- Le calcul de la matrice de covariance d'un jeu de vecteurs.

Nous nous proposons ici d'évaluer la complexité de chacune de ces opérations, puis d'en déduire la complexité algorithmique de chaque filtre de Kalman en fonction des dimensions des vecteurs d'état ( $n$ ), vecteur de mesure ( $m$ ) et de commande ( $p$ ). Pour

cette évaluation, nous considérerons comme indicateur, le nombre d'opérations à virgules flottantes effectuées.

### 2.3.1 La multiplication de matrice

Soit deux matrices  $[A]$  et  $[B]$  de dimensions respectives  $(n, m)$  et  $(m, p)$ , la multiplication  $[A][B]$  s'effectue de la façon suivante :

<p><b>Algorithme 2</b> : Multiplication de matrice</p> <pre> <b>Fonction</b> <i>Multiplie</i>(<i>A, B</i>)     <b>Entrée</b> : A matrice de dimensions (n,m)     <b>Entrée</b> : B matrice de dimensions (m,p)     <b>Sortie</b> : C matrice de dimensions (n,p)     <b>Pour</b> <i>i</i> de 0 à <i>n-1</i> faire       <b>Pour</b> <i>j</i> de 0 à <i>p-1</i> faire         somme = 0;         <b>Pour</b> <i>k</i> de 0 à <i>m-1</i> faire           somme = somme + A[<i>i, k</i>]*B[<i>k, j</i>];         <b>FinPour</b>         C[<i>i, j</i>] = somme;       <b>FinPour</b>     <b>FinPour</b> <b>FinFonction</b> </pre>
--

Avec cet algorithme, pour multiplier nos matrices nous avons :

- $n * m * p$  multiplications,
- $n * m * p$  additions,
- $n * m * (p + 1)$  affectations.

Ce qui nous donne une complexité  $T(2 * n * m * p) = \mathcal{O}(2 * n * m * p)$ .

### 2.3.2 Addition de deux vecteurs

Soit deux vecteurs  $A$  et  $B$  de longueur  $n$ , on définit le vecteur  $C = A + B$  tel que

<p><b>Algorithme 3</b> : Addition de deux vecteurs</p> <pre> <b>Fonction</b> <i>AjouterVecteur</i>(<i>A, B</i>)     <b>Entrée</b> : A Vecteur de longueur <i>n</i>     <b>Entrée</b> : B Vecteur de longueur <i>n</i>     <b>Sortie</b> : C Vecteur de longueur <i>n</i>     <b>Pour</b> <i>i</i> de 0 à <i>n - 1</i> faire       C[<i>i</i>] = A[<i>i</i>] + B[<i>i</i>];     <b>FinPour</b> <b>FinFonction</b> </pre>
---

Nous obtenons ainsi une complexité de :

- $n$  additions,
- $n$  affectations.

D'où une complexité algorithmique de  $T(n) = \mathcal{O}(n)$ .

### 2.3.3 Addition de deux matrices

Soit deux matrices  $[A]$  et  $[B]$  de dimensions  $(n, m)$ , on définit le vecteur  $[C] = [A] + [B]$  tel que

Algorithme 4 : Addition de deux matrices
<b>Fonction</b> <i>AjouterMatrices</i> ( $A, B$ )   <b>Entrée</b> : A Matrice de dimensions $(n, m)$   <b>Entrée</b> : B Matrice de dimensions $(n, m)$   <b>Sortie</b> : C Matrice de dimensions $(n, m)$   <b>Pour</b> $i$ de 0 à $n - 1$ faire     <b>Pour</b> $j$ de 0 à $m - 1$ faire       $C[i, j] = A[i, j] + B[i, j]$ ;     <b>FinPour</b>   <b>FinPour</b> <b>FinFonction</b>

Nous obtenons ainsi une complexité de :

- $n * m$  additions,
- $n * m$  affectations.

D'où une complexité algorithmique de  $T(n * m) = \mathcal{O}(n * m)$ .

### 2.3.4 Transposition de matrice

Soit une matrice  $[A]$  de dimensions  $(n * m)$ , le calcul de  $[B] = [A]^T$  se fait tel que :

Algorithme 5 : Transposition d'une matrice
<b>Fonction</b> <i>TransposeMatrice</i> ( $A$ )   <b>Entrée</b> : A Matrice de dimensions $(n, m)$   <b>Sortie</b> : B Matrice de dimensions $(m, n)$   <b>Pour</b> $i$ de 0 à $n - 1$ faire     <b>Pour</b> $j$ de 0 à $m - 1$ faire       $B[j, i] = A[i, j]$ ;     <b>FinPour</b>   <b>FinPour</b> <b>FinFonction</b>

Nous avons ici :

- $n * m$  affectations.

Avec notre indicateur basé sur les FLOPs, cette opération est considérée comme de complexité nulle. Dans les faits, on peut se passer de cette fonction en créant des fonctions annexes telles que la multiplication de matrice par la transposée d'une autre matrice.

### 2.3.5 Inversion de matrice

Selon les propriétés de la matrice, il existe plusieurs algorithmes pouvant être appliqués. Le cas le plus général est l'algorithme de Gauss-Jordan dont la complexité pour une matrice carrée de dimensions  $n * n$  est :

- $2n^3 + 2n^2$  multiplications/divisions,
- $2n^3$  soustractions,
- $2n^3 + 7n^2$  affectations.



Ce qui nous donne une complexité en FLOP de  $T(4n^3 + 2n^2) \sim \mathcal{O}(4n^3)$ .

Dans un estimateur de Kalman les matrices inversées sont des matrices de covariances Hermitiennes (et donc symétriques) qui peuvent être inversées en utilisant la décomposition de Cholesky, ce qui divise encore par deux la complexité algorithmique du calcul [10].

### 2.3.6 Calcul de vecteur moyen

Soit une matrice  $[A]$  de dimensions  $(n, m)$  composée de  $m$  vecteurs de  $n$  longueur, on calcule le vecteur moyen  $M$  tel que :

**Algorithme 6** : Calcul de vecteur moyen

```

Fonction VecteurMoyen( $A$ )
  Entrée :  $A$  Matrice composée de  $m$  vecteurs de longueur  $n$  (dimensions  $(n, m)$ )
  Sortie :  $M$  Vecteur moyen de longueur  $n$ 
  Pour  $i$  de 0 à  $n-1$  faire
    somme = 0;
    Pour  $j$  de 0 à  $m-1$  faire
      somme = somme +  $A[i, j]$ ;
    FinPour
     $M[i]$  = somme /  $m$ ;
  FinPour
FinFonction

```

On a ainsi :

- $n * m$  additions,
- $n$  divisions.

Ce qui nous donne une complexité  $T(n * (m + 1)) \sim \mathcal{O}(n * m)$ .

### 2.3.7 Calcul de matrice de covariance

Soit deux matrices  $[A]$  et  $[B]$  de dimensions respectives  $(n, m)$  et  $(p, m)$  composées de  $m$  vecteurs de longueurs respectives  $n$  et  $p$ . Nous définissons  $\bar{A}$  et  $\bar{B}$  les vecteurs moyennes de ces matrices.

Ainsi, la matrice de covariance  $[C]$  de dimensions  $(n, p)$  est définie par l'algorithme :

**Algorithme 7** : Calcul de la matrice de covariance

```

Fonction MatriceCovariance( $A, B, \bar{A}, \bar{B}$ )
  Entrée :  $A$  Matrice de dimensions  $(n, m)$  composée de  $m$  vecteurs de longueur
             $n$ 
  Entrée :  $B$  Matrice de dimensions  $(p, m)$  composée de  $m$  vecteurs de longueur  $p$ 
  Entrée :  $\bar{A}$  vecteur moyen de  $A$ 
  Entrée :  $\bar{B}$  vecteur moyen de  $B$ 
  Sortie :  $C$  Matrice de covariance de dimensions  $(n, p)$ 
  Pour  $i$  de 0 à  $n-1$  faire
    Pour  $j$  de 0 à  $p-1$  faire
      somme = 0;
      Pour  $k$  de 0 à  $m-1$  faire
        somme = somme +  $(A[i, k] - \bar{A}[i]) * (B[k, j] - \bar{B}[j])$ ;
      FinPour
       $C[i, j] =$  somme;
    FinPour
  FinPour
FinFonction

```

Ce qui nous donne les opérations :

- $n * m * p$  multiplications,
- $n * m * p$  additions,
- $n * p * (m + 1)$  affectations.

Et donc une complexité algorithmique de  $T(2 * n * m * p) = \mathcal{O}(2 * n * m * p)$ .

### 2.3.8 Récapitulatif

En résumé, les complexités de calcul de chaque opération mathématique peut se répertorier suivant la Table 2.2.

Opération	$T(\cdot)$	$\mathcal{O}(\cdot)$
Multiplication de matrice	$2 * n * m * p$	$2 * n * m * p$
Addition de deux vecteurs	$n$	$n$
Addition de deux matrices	$n * m$	$n * m$
Transposition de matrice	0	0
Inversion de matrice	$4 * n^3 + 2 * n^2$	$4 * n^3$
Vecteur moyen d'une matrice	$n * (m + 1)$	$n * m$
Moyenne d'un vecteur	$n + 1$	$n$
Covariance de deux matrices	$2 * n * m * p$	$2 * n * m * p$

TABLE 2.2 – Récapitulatif des complexités algorithmiques des opérations mathématiques

## 2.4 Complexité du filtre de Kalman

À partir de là, nous pouvons estimer la complexité des filtres de Kalman, EKF et UKF en fonction des longueurs de leurs vecteurs d'état  $n$ , de mesure  $m$  et de commande  $p$ , pour

chaque étape de calcul (*c.f.* Table 2.3) et pour l'algorithme complet (*c.f.* Table 2.4).

Algo	Opération	$\mathcal{O}(\cdot)$
(E)KF	$\tilde{\mathbf{X}}_{k+1} = [\mathbf{A}]\hat{\mathbf{X}}_k + [\mathbf{B}]\mathbf{U}_k$	$2n^2$
	$[\tilde{\mathbf{P}}_{k+1}] = [\mathbf{A}][\hat{\mathbf{P}}_k][\mathbf{A}]^T + [\mathbf{Q}]$	$4n^3$
	$\tilde{\mathbf{Y}}_k = [\mathbf{C}]\tilde{\mathbf{X}}_k + [\mathbf{D}]\mathbf{U}_k$	$2m(n+p)$
	$\mathbf{E}_k = \mathbf{Y}_k - \tilde{\mathbf{Y}}_k$	$m$
	$[\mathbf{K}_{k+1}] = [\tilde{\mathbf{P}}_k][\mathbf{C}]^T([\mathbf{C}][\tilde{\mathbf{P}}_k][\mathbf{C}]^T + [\mathbf{R}])^{-1}$	$4n^2m / 4m^2n$
	$\hat{\mathbf{X}}_k = \tilde{\mathbf{X}}_k + [\mathbf{K}_k]\mathbf{E}_k$	$2mn$
	$[\hat{\mathbf{P}}_k] = (\mathbf{I} - [\mathbf{K}_k][\mathbf{C}])[\tilde{\mathbf{P}}_k]$	$\sim 2n^3 / 2m^2n$
UKF	$\mathcal{X}_k = \left[ \hat{\mathbf{X}}_k, \hat{\mathbf{X}}_k \pm \sqrt{(n+\lambda)[\hat{\mathbf{P}}_k]} \right]$	$n^3$
	$\tilde{\mathcal{X}}_{k+1}^{(i)} = f(\mathcal{X}_k^{(i)})$	$2n\mathcal{O}(f(\cdot))$
	$\tilde{\mathbf{X}}_{k+1} = \sum_{i=0}^{2n} \omega_i^\mu \tilde{\mathcal{X}}_{k+1}^{(i)}$	$4n^2$
	$[\tilde{\mathbf{P}}_{k+1}] = \sum_{i=0}^{2n} \omega_i^c (\tilde{\mathcal{X}}_{k+1}^{(i)} - \tilde{\mathbf{X}}_{k+1})(\tilde{\mathcal{X}}_{k+1}^{(i)} - \tilde{\mathbf{X}}_{k+1})^T + [\mathbf{Q}]$	$6n^3$
	$\tilde{\mathcal{Y}}_{k+1}^{(i)} = g(\tilde{\mathcal{X}}_{k+1}^{(i)}, \mathbf{U}_{k+1}, 0)$	$(2n+1)\mathcal{O}(g(\cdot))$
	$\tilde{\mathbf{Y}}_{k+1} = \sum_{i=0}^{2n} \omega_i^\mu \tilde{\mathcal{Y}}_{k+1}^{(i)}$	$4m^2n$
	$[\tilde{\mathbf{P}}_{yy,k+1}] = \sum_{i=0}^{2n} \omega_i^c (\tilde{\mathcal{Y}}_{k+1}^{(i)} - \tilde{\mathbf{Y}}_{k+1})(\tilde{\mathcal{Y}}_{k+1}^{(i)} - \tilde{\mathbf{Y}}_{k+1})^T + [\mathbf{R}]$	$6m^2n$
	$[\tilde{\mathbf{P}}_{xy,k+1}] = \sum_{i=0}^{2n} \omega_i^c (\tilde{\mathcal{X}}_{k+1}^{(i)} - \tilde{\mathbf{X}}_{k+1})(\tilde{\mathcal{Y}}_{k+1}^{(i)} - \tilde{\mathbf{Y}}_{k+1})^T$	$4n^2m$
	$\hat{\mathbf{X}}_k = \tilde{\mathbf{X}}_k + [\mathbf{K}_k]\mathbf{E}_k$	$2mn$
	$[\hat{\mathbf{P}}_k] = (\mathbf{I} - [\mathbf{K}_k][\mathbf{C}])[\tilde{\mathbf{P}}_k]$	$\sim 2n^3 / 2m^2n$

TABLE 2.3 – Complexité des étapes de filtrage en fonction de  $n$ ,  $m$  et  $p$

Algorithme	$T(\cdot)$	$\mathcal{O}(\cdot)$
(E)KF	$4n^3 + 4m^3 + 6m^2n + 4n^2m + 3n^2 + \dots$	$4n^3$
UKF	$10n^3 + 4n^2m + 14m^2n + 23n^2 + 6m^2 + \dots$	$10n^3$

TABLE 2.4 – Complexité des filtres de Kalman

Nous remarquons ainsi que bien que les filtres EKF et UKF appartiennent à la même classe de complexité ( $\sim \mathcal{O}(n^3)$ ), l'algorithme UKF est environ deux fois plus demandeur en termes de temps de calcul.

## 2.5 Diminution de la complexité des estimateurs

Nous venons de voir que la complexité des algorithmes pouvait avoir un impact non négligeable sur la faisabilité d'un capteur temps-réel. Ainsi, l'une des solutions les plus communes pour permettre la faisabilité sans augmenter les coûts de production consiste à réaliser des optimisations algorithmiques, ou à diminuer la consommation de ressources des solutions mises en place. Nous allons explorer au cours de cette section les solutions permettant de résoudre ce problème.

Estimateur	Complexité du système complet (FLOPs)	Complexité factorisée (FLOPs)
EKF	8788	2548
UKF	21970	6370
KF + UKF	N/A	3298

TABLE 2.5 – Comparaison de complexité entre un système factorisé ou complet (pour 13 états)

### 2.5.1 Factorisation d'états

Nous avons vu sur la Table 2.4 que la complexité du filtre de Kalman était principalement influencée par la dimension de son vecteur d'état et proportionnelle à  $\mathcal{O}(n^3)$ . Ainsi, la diminution du nombre d'états  $n$ , ou la séparation du système en plusieurs sous-systèmes indépendants permet de nettement diminuer la charge de calculs.

De plus, dans la plupart des cas, la dynamique thermique est modélisée sous forme linéaire et constitue un sous système influant les mesures. Par exemple, si nous prenons le système UQS discuté en Chapitre 3, le vecteur d'état complet du système varie de 13 à 14 états. Cependant, il peut être factorisé en un système linéaire de 8 états et un système non-linéaire de 5 à 6 états, ce qui nous donne une complexité en FLOP selon le tableau 2.5.

Nous remarquons ainsi que par factorisation du système en une partie linéaire et une partie non linéaire, on obtient une diminution de complexité d'un facteur 6.

### 2.5.2 Observateur à gain commuté

#### Approximation du modèle

Parfois, le modèle non-linéaire du système, ou le calcul de ses matrices jacobiniennes peuvent être longs à itérer pour un système embarqué. On est alors amené à simplifier ce système en approximant son fonctionnement à une ou plusieurs fonctions simples à itérer. On peut par exemple faire une régression sur un jeu de données important pour en tirer un polynôme de faible degré, moins précis, mais permettant au programme de rentrer dans la mémoire du micro-contrôleur.

#### LUT

Dans le cas de fonctions représentatives du système non identifiables ou présentant des comportements difficiles à calculer (racines, trigonométrie...), l'utilisation de tables de recherche peut être mise en place avec une interpolation simple entre les points. Au travers de cette technique, il est possible de faire des calculs de fonctions multi-dimensionnelles complexes avec un petit micro-contrôleur, pour peu que la mémoire disponible soit suffisante pour stocker les tableaux de points.

#### Observateur de Kalman statique

La principale difficulté de l'algorithme du filtre de Kalman réside dans les diverses multiplications et l'inversion de matrice liée au calcul du gain dynamique de correction. L'utilisation de la technique des LUT appliquée au gain de Kalman permet de fortement

simplifier la complexité du filtre [11]. On peut par exemple pré-calculer des matrices de gain que nous stockerons en mémoire et que nous sélectionnerons en fonction de l'état du système. Cette solution réduit le filtre de Kalman à un simple observateur dit à gain commuté.

L'algorithme se résume alors aux étapes et complexités présentées en table 2.6.

Algo	Opération	$\mathcal{O}(\cdot)$
Gain commuté	$\tilde{\mathbf{X}}_{k+1} = [\mathbf{A}]\hat{\mathbf{X}}_k + [\mathbf{B}]\mathbf{U}_k$	$2n^2$
	$\tilde{\mathbf{Y}}_k = [\mathbf{C}]\tilde{\mathbf{X}}_k + [\mathbf{D}]\mathbf{U}_k$	$2m(n+p)$
	$\mathbf{E}_k = \mathbf{Y}_k - \tilde{\mathbf{Y}}_k$	$m$
	$\hat{\mathbf{X}}_k = \tilde{\mathbf{X}}_k + [\mathbf{K}_k]\mathbf{E}_k$	$2mn$

TABLE 2.6 – Complexité des étapes de filtrage à gain commuté en fonction de  $n$ ,  $m$  et  $p$

Ainsi, la complexité totale d'un tel filtre diminue pour devenir  $\mathcal{T}(\cdot) = 4 * m * n + 2 * m * p + 2 * m + 2 * n^2 + 2 * n * p + 2 * n$ , soit  $\mathcal{O}(\cdot) = 2 * n^2$ .

### 2.5.3 Calculs en virgule fixe

Comme évoqués précédemment, les micro-contrôleurs généralement utilisés dans les capteurs sont dépourvus d'unités de calculs flottants et doivent donc effectuer toutes les opérations relatives à ces calculs par émulation logicielle. Ainsi, le temps consacré à une opération flottante est nettement plus important que celui consacré à la même opération sur un entier.

La solution d'optimisation proposée consiste donc à réaliser tous les calculs en entier avec un facteur d'échelle prédéterminé, et correspondant à l'application (permettant le meilleur compromis précision/plage maximum de mesure) [12]. Bien que très performante, cette solution impose une étude préalable rigoureuse pour la sélection de la représentation et peut provoquer des instabilités numériques sur certaines opérations (par exemple l'inversion d'une matrice de covariance). Elle est donc à ne réserver que pour les cas nécessitant les optimisations les plus extrêmes.

### 2.5.4 Pre-processing des mesures pour un système linéaire

Dans certains cas, les mesures réalisées par un capteur peuvent être simplement traduites en informations linéaires. Cette traduction peut être effectuée par LUT ou par un calcul peu coûteux, ce qui simplifie fortement l'estimateur en aval.

Par exemple, dans le cas de notre capteur de rayonnement solaire évoqué au chapitre 1.2, il est possible de remplacer la mesure de CTN dans l'équation du modèle par la température correspondante à cette mesure ( $T_s$ ) et ainsi revenir à un filtre de Kalman linéaire, plus simple à calculer.

Il faut cependant noter que dans ce cas, la non-linéarité du bruit de mesure n'est pas répercuté sur le filtre, ce qui peut conduire à une solution sous-optimale.

## 2.6 Conclusion

Au cours de ce chapitre, nous avons vu que le fait d'avoir un algorithme fonctionnel hors ligne, à puissance de calcul illimité ne garantissait pas la possibilité de réaliser un

capteur intelligent temps-réel. Nous avons parcouru un ensemble de techniques permettant d'étudier, puis de diminuer la complexité des algorithmes exécutés par le micro-contrôleur afin de rendre réalisable le système.

Au cours du prochain chapitre, nous allons mettre en oeuvre la méthodologie que nous venons de développer sur une application de capteurs intelligents embarqués : le projet UQS utilisant un système de spectroscopie infrarouge embarqué et de mesure de températures pour estimer la concentration d'urée en solution aqueuse dans un réservoir d'*AdBlue*, pour la dépollution des gaz d'échappement sur les véhicules à moteur.

Nous nous intéresserons ensuite à l'application de qualification du geste sportif, abordée par le projet *Architecture Reconfigurable Embarquée Multi-capteurs* (AREM), qui présente une contrainte temps-réel forte avec un besoin de traitements de données à une fréquence de 50Hz.

L'étude de ces deux capteurs présente des intérêts variés de par les différentes échelles de temps et de compensation qu'ils abordent, en pointant cependant vers un même but : traduire des données d'entrées multi-physiques complexes vers un indicateur simple.

“L'ère de la sobriété, de la simplicité heureuse a sonné.”, *Pierre RABHI*



# Bibliographie

- [1] James Martin. *Programming real-time computer systems*. Prentice-Hall, 1965.
- [2] Robert B. Miller. Response time in man-computer conversational transactions. *International Business Machines Corporation*, 1968.
- [3] K.G. Shin ; P. Ramanathan. Real-time computing : a new discipline of computer science and engineering. *Proceedings of the IEEE*, 82 :6–24, January 1994.
- [4] M.E.M.B. Gaid ; A. Cela ; Y. Hamam. Optimal integrated control and scheduling of networked control systems with communication constraints : application to a car suspension system. *IEEE Transactions on Control Systems Technology*, 14 :776–787, 2006.
- [5] IC Insights. The mclean report, 2014.
- [6] Jean-Pierre Deschamps ; Géry Jean Antoine Bioul ; Gustavo D. Sutter. *Synthesis of Arithmetic Circuits : FPGA, ASIC, and Embedded Systems*. Wiley Online Library, Mars 2006.
- [7] Sameh Galal ; Mark Horowitz. Energy-efficient floating-point unit design. *IEEE Transactions on Computers*, 60 :913–922, 2010.
- [8] STMicroelectronics. Floating point unit demonstration on stm32. An4044, Application Note, May 30 2016.
- [9] S. Russell and P. Norvig. *Intelligence artificielle : Avec plus de 500 exercices*. Pearson Education. Pearson, 2010.
- [10] G.H. Golub ; C.F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, 3rd edition, 1996.
- [11] B. Sinopoli ; L. Schenato ; M. Franceschetti. Kalman filtering with intermittent observations. *IEEE Transactions on Automatic Control*, 49 :1453–1464, 2004.
- [12] S. Ardalan ; S. Alexander. Fixed-point roundoff error analysis of the exponentially windowed rls algorithm for time-varying systems. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 35 :770–783, 2003.





# Chapitre 3

## Un cas pratique industriel : le projet UQS

Au cours des chapitres précédents nous avons décrit une méthodologie que nous avons utilisée pour réaliser des capteurs embarqués intelligents. Nous avons d'abord exploré le point de vue modélisation, représentation et traitement du signal, puis le point de vue contraintes d'implémentation liées au matériel utilisé.

Nous allons maintenant nous intéresser à l'application de cette méthode sur un cas pratique : le capteur de qualité d'urée UQS (*Urea Quality Sensor*) développé et commercialisé par *MEAS-France / TE Connectivity*.

Après une description de son principe de fonctionnement et des informations fournies par le capteur, nous listerons les paramètres d'influence pouvant agir sur les différents éléments du système. Nous nous intéresserons ensuite aux modèles multi-physiques que nous avons dressés du système et de ses éléments constitutifs afin de décrire au mieux son comportement en fonction des paramètres d'influence. Puis, nous présenterons les différentes représentations que nous avons élaborées du système et en étudierons l'observabilité, avant de parler des différents filtres et observateurs découlants de ces représentations ainsi que de leurs performances.

Finalement, nous aborderons les problématiques liées à l'implémentation sur la cible réelle embarquée, avant de présenter les solutions proposées.

### 3.1 Principe de fonctionnement

Le capteur UQS est un capteur embarqué, développé et commercialisé par *MEAS-France / TE Connectivity*, chargé de mesurer la concentration d'urée en solution aqueuse dans un réservoir. Ce capteur est utilisé dans le cadre du système SCR (*Selective Catalytic Reduction*) pour la dépollution des oxydes d'azote ( $NO$ ,  $NO_2$ ) sur les gros moteurs diesel [1][2]. Il utilise le principe de la spectroscopie proche infra-rouge (*Near Infra-Red*, NIR) pour déterminer le type de fluide dans le réservoir et, dans le cas de l'urée, mesurer sa concentration.

Les travaux effectués autour de la mesure optique d'ammoniac au sein du système SCR portent actuellement principalement sur sa mesure en phase gazeuse [3][4].

### 3.1.1 Spectroscopie NIR

Le principe sur lequel s'appuie le capteur UQS est la mesure de l'absorption électromagnétique d'une substance dans le proche infra-rouge pour en déterminer sa composition. Ce principe, nommé spectroscopie, est utilisé depuis de nombreuses années en laboratoire pour l'analyse d'espèces chimiques [5].

Ainsi, pour effectuer cette mesure, un spectromètre de laboratoire peut émettre une lumière blanche, et lorsqu'il est étalonné sur celle-ci, mesurer la modification du spectre lorsque l'on intercale un échantillon entre la source lumineuse et l'élément sensible (Figure 3.1). Dans ce cas, nous obtenons le spectre d'absorption de cette espèce aux longueurs d'ondes mesurées, sous la forme de counts (unité élémentaire de sortie des convertisseurs analogique/numériques), dépendant du paramétrage et de la calibration du spectroscope.

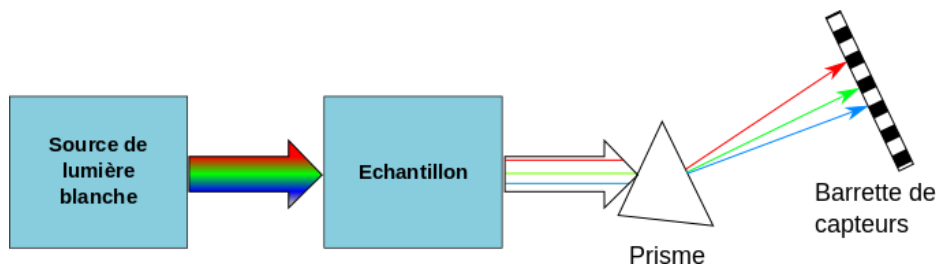


FIGURE 3.1 – Principe de fonctionnement d'un spectromètre

La mesure dans le NIR (bande des 750nm à 1100nm dans notre cas) permet de déterminer la présence d'espèces chimiques telles que l'eau et l'urée [6] (*c.f.* Figure 3.2). Dans ce cas, afin de s'affranchir du paramétrage du spectroscope, nous avons normalisé les courbes en ratio par rapport à la valeur maximale du spectre de l'urée à 25°C.

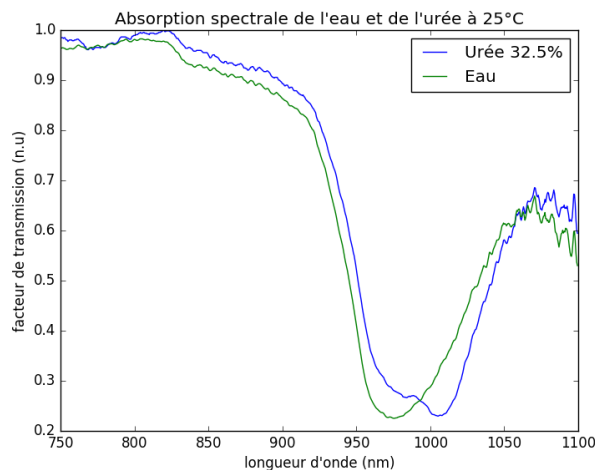


FIGURE 3.2 – Spectres d'absorption de l'eau et de l'urée à 32.5% sur 3cm à 25°C

Il est ainsi possible de déterminer à partir d'un spectromètre de laboratoire la concentration d'urée en solution aqueuse par analyse du spectre d'absorption du liquide. Pour ce projet, nous faisons l'hypothèse que le spectre d'absorption évolue linéairement pour chaque longueur d'onde en fonction de la concentration en urée, et peut donc être complètement déterminé à partir des mesures d'absorption réalisées.

La spectroscopie de laboratoire impose cependant des contraintes importantes quant aux conditions de mesure. En effet, la source lumineuse ainsi que le capteur doivent

être étalonnés avant chaque utilisation et maintenus dans un environnement contrôlé (température, humidité, poussière...). Le processus de mesure impose donc un matériel volumineux, coûteux et difficile à mettre en oeuvre qui ne peut être embarqué tel quel sur un véhicule.

Afin de simplifier le fonctionnement et de diminuer les coûts, tout en permettant au capteur de fonctionner dans une plage de température plus large, le capteur UQS repose sur la sélection de trois points du spectre et l'utilisation de composants optroniques standards pour réaliser sa mesure.

### 3.1.2 Le capteur UQS

Le capteur UQS est un capteur devant mesurer la concentration d'urée sur un véhicule et transmettre cette information au calculateur central au travers du bus CAN. Il doit donc être capable d'effectuer les mesures en continu dans un environnement défavorable soumis à de fortes variations environnementales et à des vibrations, le tout en garantissant une fiabilité de fonctionnement et des coûts de productions compatibles avec le secteur automobile.

Pour répondre à ce besoin, l'architecture de laboratoire spectromètre/PC a été simplifiée en se basant sur trois sources lumineuses à spectre étroit sélectionnées en fonction du besoin spécifique, de capteurs photo-diodes à large spectre de sensibilité et d'une architecture basée sur le micro-contrôleur 8bits (Figure 3.3).

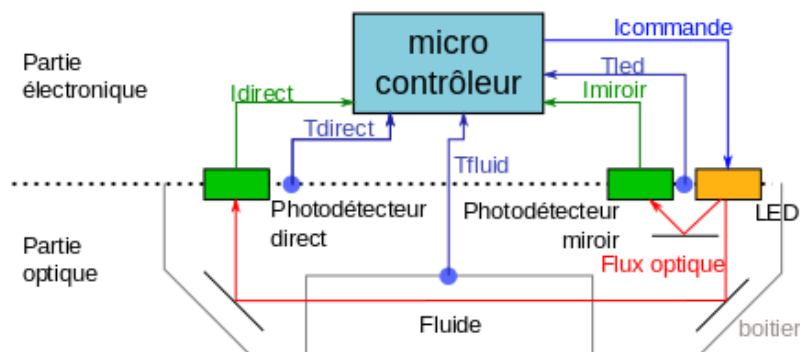


FIGURE 3.3 – Architecture matérielle du capteur UQS

Ainsi les composants sélectionnés pour effectuer l'émission optique sont des LEDs (*Light Emitting Diodes*) qui sont des composants présentant un spectre d'émission optique relativement fin. La réception quant à elle est effectuée par des photo-diodes dont la sensibilité spectrale est concentrée dans le NIR [7].

Afin de choisir les longueurs d'ondes optimales pour effectuer les mesures, nous pouvons construire la Figure 3.4 par soustraction du spectre de l'eau à celui de l'urée.

Nous choisissons donc de réaliser trois points de mesure sur cette courbe : un point de sensibilité nulle à la concentration, utilisé comme référence, et deux points sensibles à la concentration. Les longueurs d'onde sélectionnées sont  $810nm$ ,  $970nm$  et  $1020nm$ .

Nous obtenons ainsi pour les trois LEDs les spectres d'émission suivants (Figure 3.5).

Contrairement aux spectromètres de laboratoires qui émettent une lumière blanche et décomposent le spectre à la réception, ce qui leur permet en une mesure de connaître le spectre d'absorption de l'échantillon mesuré (voir Figure 3.1), le produit UQS utilise deux photo-diodes : une pour la voie directe, permettant d'effectuer les mesures d'absorption, et une sur la voie miroir permettant de compenser les variations de caractéristiques des

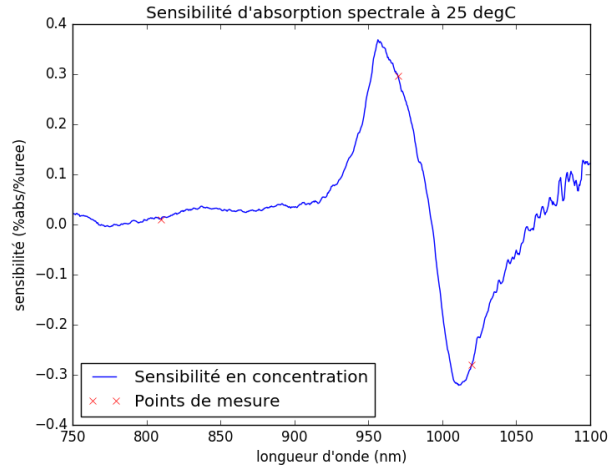


FIGURE 3.4 – Sensibilité spectrale de la concentration d’urée à 25°C

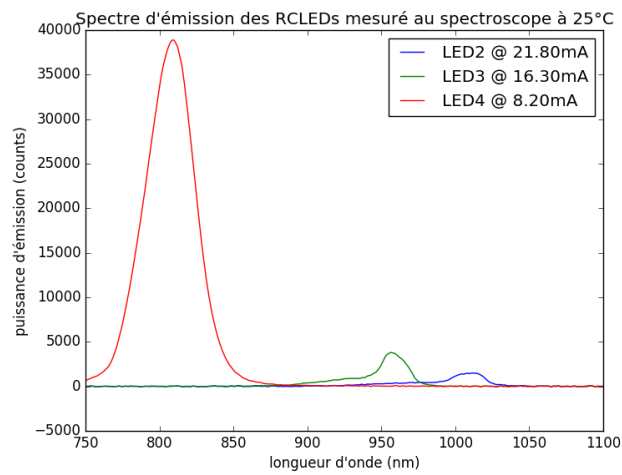


FIGURE 3.5 – Spectre d’émission des LEDs à 25°C  
 $S_{led,i}(\lambda, 25, L_i, current)$

LEDs dans le temps (Figure 3.6). Aucun de ces détecteurs n’étant sélectif en longueur d’onde (Figure 3.7), l’analyse spectrale est effectuée par allumage et mesure séquentielle de chacune des LEDs (Figure 3.8).

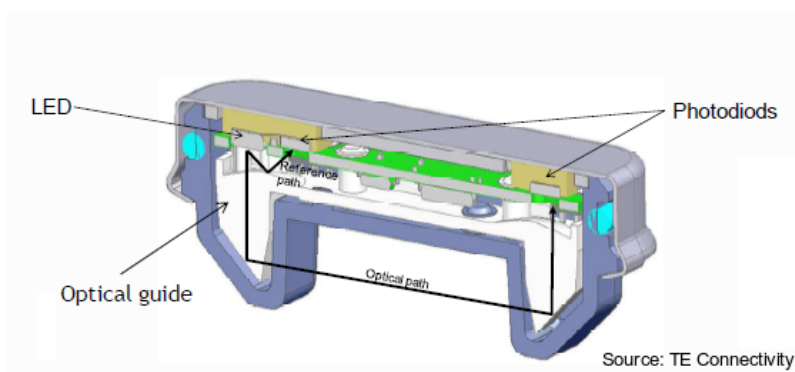


FIGURE 3.6 – Principe de fonctionnement du capteur UQS

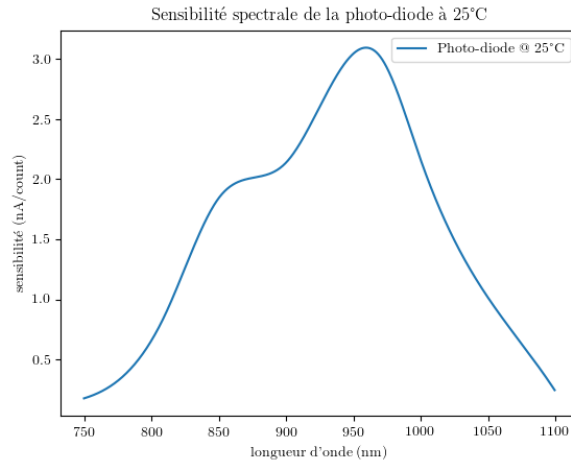


FIGURE 3.7 – Sensibilité spectrale des photo-diodes à 25°C  
 $sensibilite(\lambda, 25)$

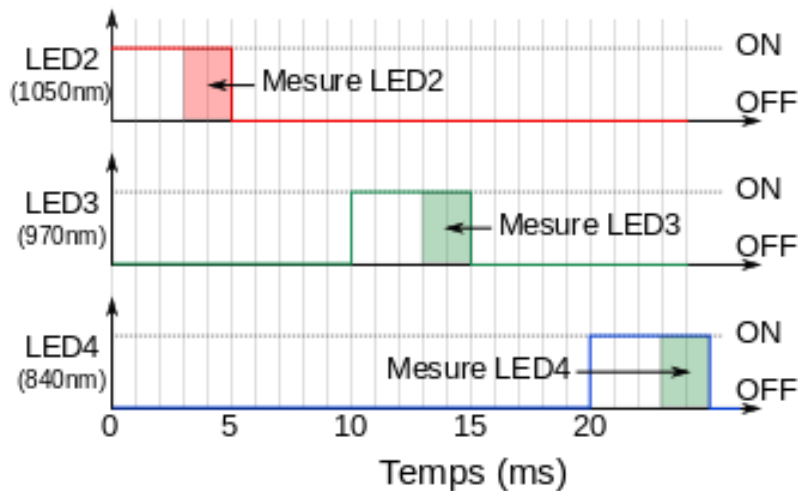


FIGURE 3.8 – Séquence d’allumage des LEDs

### 3.1.3 Grandeurs d’entrées-sorties

D’un point de vue système optique, nous pouvons considérer deux grandes catégories d’entrées :

- les entrées commandées par la partie intelligente du système : les courants d’excitation des LEDs dans notre cas,
- les entrées de mesures (ou grandeurs d’influence) du système : ici, il s’agira des températures des différents éléments, ainsi que du type et de la concentration du fluide.

Les sorties quant à elles seront les valeurs mesurables par l’unité numérique de traitement, à savoir les températures des éléments (prises à proximité des LEDs, photo-diodes et du chemin optique) ainsi que les courants en sortie des photo-diodes.

## 3.2 Paramètres d'influence et modèle

Les conditions de fonctionnement du produit, immergé dans un réservoir sur un véhicule, peuvent fortement varier au cours de sa vie. Il peut ainsi voir :

- des variations de température importantes (selon les saisons, l'exposition ou les utilisations, le produit peut fonctionner entre  $-10^{\circ}C$  et  $+80^{\circ}C$ , avec parfois des évolutions thermiques rapides lors de l'activation du circuit de dégivrage de l'urée),
- des fluides inappropriés, lorsque l'opérateur de maintenance du véhicule effectue, par erreur ou à propos, le remplissage avec le mauvais fluide (gasoil, eau, lave glace...),
- des altérations du chemin optique, lorsque le réservoir est sale (contient de la poussière) ou que l'urée dégaze (produit des bulles qui peuvent se fixer le long du chemin optique),
- des vibrations liées au déplacement du véhicule...

### 3.2.1 La thermique au coeur du système

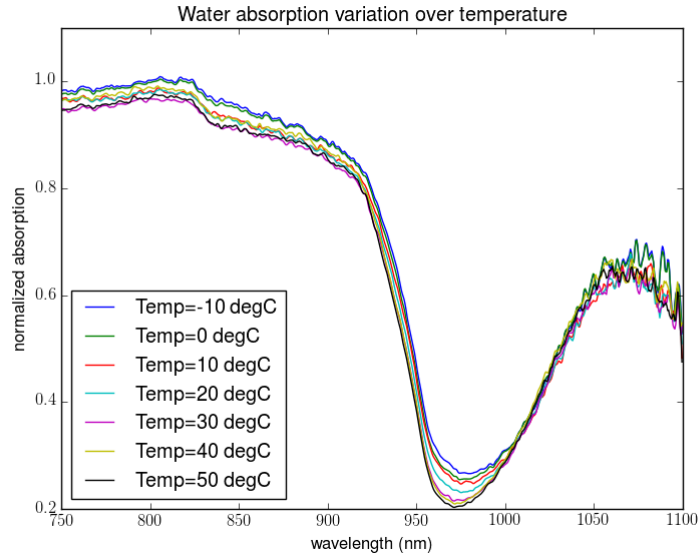
De par son architecture, le système présente une forte sensibilité à la température de ses différents éléments. Nous pouvons par exemple remarquer que les spectres d'absorption de l'eau et de l'urée varient en température (Figures 3.9a et 3.9b)[8], ce qui impose une mesure de la température du fluide. De plus, les comportements thermiques des composants électroniques et optroniques (LEDs, photo-diodes, amplificateurs...) sont eux aussi fortement sensibles à chacune de leurs températures [9] (Figures 3.10a et 3.10b).

### 3.2.2 Obstructions optiques

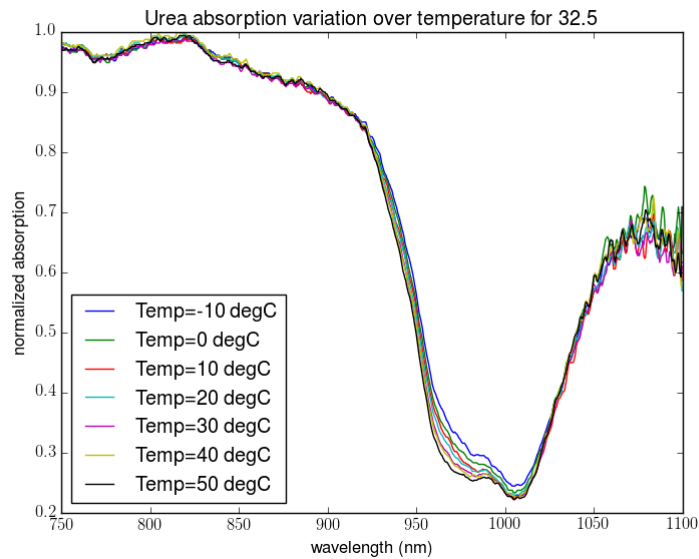
Le principe de fonctionnement du capteur, se basant sur le comportement optique du fluide, est de fait sensible aux événements se produisant sur ce chemin que nous pourrions classifier selon 4 catégories :

- **Obstruction partielle, identique sur toutes les voies** : il s'agit d'une atténuation du signal de façon proportionnelle sur toutes les voies. Ce cas de figure peut être facilement observé grâce à la LED4 qui n'est pas sensible à la concentration et à la redondance des mesures,
- **Obstruction partielle, sur une ou plusieurs voies de façon décorrélée** : il peut par exemple s'agir d'une bulle ou de poussière dans le chemin optique. Ce cas est plus difficile à traiter que le précédent car il peut affecter le principe même de fonctionnement du capteur s'il n'est pas diagnostiqué. Une redondance des LEDs en plusieurs positions permet de diminuer l'impact du problème,
- **Obstruction totale sur toutes les voies** : il peut s'agir d'un fluide givré ou d'un obstacle dans le chemin optique. Aucune mesure ne peut être effectuée dans ces conditions,
- **Obstruction totale sur une ou plusieurs voies** : Dans le cas d'une particule opaque dans le chemin optique d'une ou plusieurs LEDs. Cette erreur est généralement facilement détectée et l'obstruction de courte durée, ce qui permet de continuer la mesure par élimination des points défailants.

Au cours de notre étude, nous ne nous occuperons donc que du cas d'une obstruction partielle, identique ou non sur toutes les voies, mais évoluant de façon lente (par exemple dans le cas de dégradation des propriétés optiques des plastiques au cours des mois d'utili-



(a) sensibilité thermique de l'eau :  $absorption(\lambda, 0, T)$



(b) sensibilité thermique de l'urée :  $absorption(\lambda, 32.5, T)$

FIGURE 3.9 – Sensibilité des spectres d'absorption à la température du fluide

sation). Les cas d'obstructions totales ou d'obstructions rapides seront considérés comme étant traités en amont par un autre algorithme travaillant sur les données brutes.

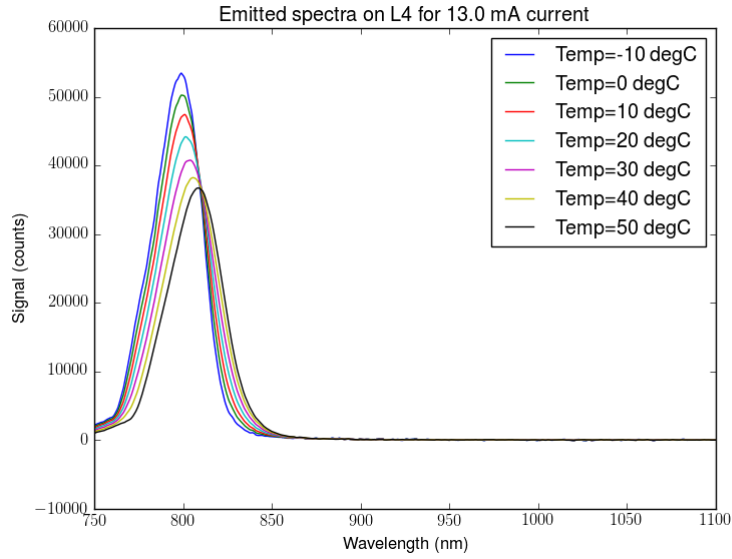
### 3.2.3 Sensibilité mécanique

L'utilisation de composants optroniques, et en particulier de LEDs mises en boîtiers compacts (Figure 3.11), donne au système une sensibilité mécanique liée à l'alignement des éléments optiques.

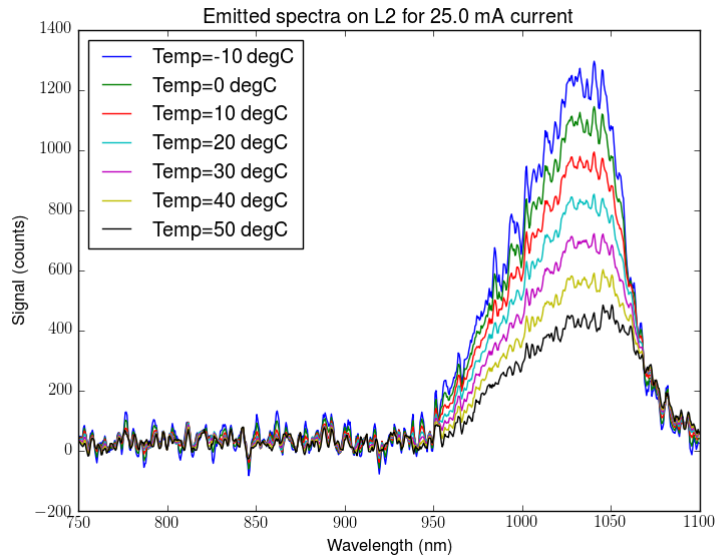
Cette sensibilité peut se traduire de deux façons différentes :

- par une dispersion des puissances reçues d'un produit à l'autre,
- par une sensibilité thermique des gains de transmission optique dues aux dilatations





(a) LED4 :  $S_{led4}(\lambda, T, L4_{current})$



(b) LED2 :  $S_{led2}(\lambda, T, L2_{current})$

FIGURE 3.10 – Sensibilité thermique des LEDs

des différents éléments mécaniques et optiques du capteur en fonction de leurs températures.

La dispersion de centrage d'un capteur à l'autre sera un élément de calibration que nous modéliserons par des gains directs et miroirs. Le comportement thermique quant à lui, devra être inclus de façon dynamique et dépendante de la température.

### 3.3 Approche orientée modèle

Afin d'effectuer la mesure de la concentration d'urée à partir du capteur UQS, notre première approche a consisté à dresser un modèle multi-physique du capteur et de ses

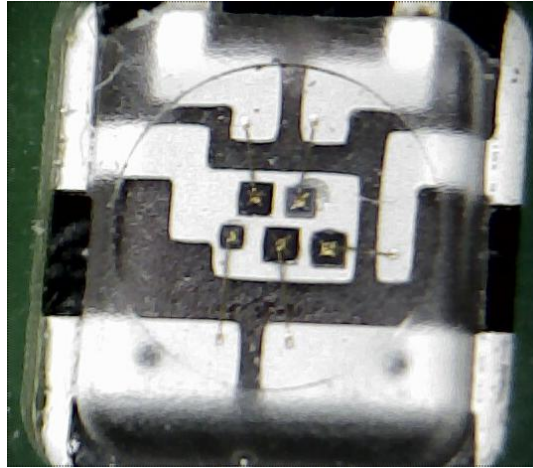


FIGURE 3.11 – LED package UQS

éléments et d'en déterminer les paramètres d'influence :

- Les paramètres de dispersion, qui décrivent la spécificité d'un capteur par rapport à un autre,
- Les paramètres de vieillissement, qui décrivent les modifications du comportement du capteur au cours de sa vie.

### 3.3.1 Paramètres de dispersion

Parmi les paramètres de dispersion, nous retrouvons la dispersion intrinsèque à tous les composants utilisés, qu'il s'agisse de tolérances mécaniques, de dispersions de résistances, ou de tout autre composant électronique. Nous considérerons cependant que cette dispersion se répartit majoritairement selon deux axes :

- **La dispersion des spectres d'émission des LEDs**, principalement par rapport à leur longueur d'onde centrale, qui détermine leur position vis-à-vis du spectre d'absorption des fluides mesurés.
- **L'influence du montage mécanique du capteur sur les gains** de transmission, de réflexion optique ainsi que sur la sensibilité thermique due à la dilatation des matériaux.

Ces paramètres de dispersion seront compensés dans le modèle du capteur lors de la calibration en usine de celui-ci ou lors de la première utilisation.

### 3.3.2 Paramètres de vieillissement

Comme tout système physique, le capteur est soumis à un processus de vieillissement qui altère son comportement. Dans le cas de capteurs en laboratoire, cette dérive est compensée par un ré-étalonnage régulier, voire systématique avant chaque utilisation du système. Pour un capteur embarqué sur véhicule, nous ne pouvons cependant pas nous permettre d'utiliser une procédure aussi lourde, ceci sera donc compensé par l'utilisation de paramètres liés au vieillissement qui seront remis à jour de façon régulière par le système lorsque les conditions seront stables. Les principaux impacts du vieillissement étant :

- **Une diminution lente de la puissance optique émise par les LEDs pour une puissance électrique donnée en entrée (diminution du rendement)**

[10] : cette dérive est observable car affectant à la fois la voie directe et la voie miroir de façon proportionnelle, et donc aisément compensable,

- **Une opacification des plastiques sur le chemin optique liée à leur vieillissement** [11] : cette dérive peut être délicate à observer car n’affectant pas de façon identique toutes les longueurs d’onde. Un réajustement des paramètres du modèle en conditions connues (ré-étalonnage) peut être nécessaire pour corriger ce vieillissement,
- **Une humidification du boîtier, liée à la pénétration d’eau due à l’immersion continue dans le fluide** : cette humidification modifiant le comportement optique, voire le rendant inopérant en cas de condensation, ne sera pas adressée par les algorithmes traités ici.

### 3.3.3 Modèle des éléments optiques

A partir de ces paramètres, nous avons dressé les modèles multi-physiques et multi-paramétriques des trois principaux constituants de la chaîne de mesure, à savoir :

- Les LEDs, sensibles aux courants d’excitation et à la température, ces dernières fournissent un spectre optique,
- Les photo-diodes, sensibles à un spectre optique et à la température, ils fournissent en sortie un courant,
- le fluide le long du trajet optique (nous ne nous concentrons que sur l’urée dans notre cas), sensible à la température et à la concentration d’urée, celui-ci absorbe et modifie le spectre optique qui le parcourt.

Nous allons détailler dans les trois prochaines parties les modèles dont nous avons tiré ces éléments.

#### LEDs

Nous pouvons représenter une LED selon le schéma bloc en Figure 3.12.

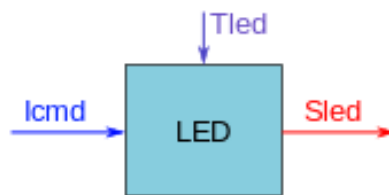


FIGURE 3.12 – Schéma bloc d’une LED  
 $S_{led,i}(\lambda, T_{led}, I_{cmd})$

Cela signifie qu’une LED est un composant commandé par un courant  $I_{cmd}$ , subissant l’impact d’une température  $T_{led}$  et émettant un spectre  $S_{led}$ . Le modèle que nous devons réaliser aura donc deux paramètres physiques d’entrée et une sortie.

Pour que le modèle soit représentatif, nous devons donc établir la relation permettant de déterminer le spectre de chacune des trois LED en fonction des variations des courants de commande et des températures.

Ces modèles ont été réalisés sous la forme de tables LUT à 2 entrées, renseignées par les données mesurées en laboratoire sur chaque LED. Ces données nous ont été fournies par *MEAS-France / TE Connectivity* qui les a collectées selon la procédure suivante :

- Le boîtier LED est fixé sur un module à effet Peltier permettant d'en asservir la température, et le flux optique de sortie est transmis à un spectroscope (Figure 3.13),
- Le module à effet Peltier est réglé de façon successive pour que la température des LEDs soit de 0, 20, 40 et 60°C,
- Pour chaque température, le courant d'excitation de chaque LED est piloté pour des valeurs allant de 8 à 50mA selon les LEDs,
- Pour chaque couple température/courant, le spectre de chaque LED est mesuré et sauvegardé dans un fichier séparé.

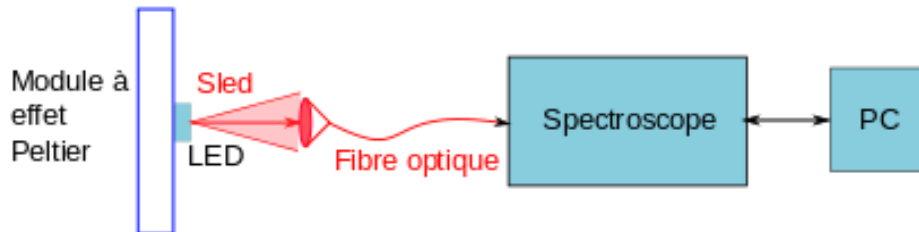


FIGURE 3.13 – Manipulation de caractérisation des LEDs

A partir de ces mesures, le modèle que nous utilisons consiste en une interpolation linéaire du spectre de sortie entre ces points de caractérisation, en fonction de la température et du courant auxquels sont soumis les LEDs (Figure 3.10).

## Photo-diodes

La photo-diode quant à elle fonctionne comme la réciproque de la LED : à un spectre d'entrée et à une température donnés, il fait correspondre un courant de sortie (Figure 3.14).



FIGURE 3.14 – Schéma bloc de la photo-diode

$$I_{det}(T_{det}, S_{entree})$$

Pour modéliser ce capteur, nous considérons que son courant de sortie est la somme de :

- son courant d'obscurité ( $I_{dark}$ ), dépendant uniquement de la température et mesurable lorsque le capteur est dans le noir total (son spectre d'entrée est nul quelle que soit la longueur d'onde),
- la somme des contributions de chaque puissance spectrale reçue multipliée par la sensibilité spectrale correspondante (équation 3.1).

$$I_{det}(T_{det}, S_{entree}) = I_{dark}(T_{det}) + \int_{-\infty}^{+\infty} S_{entree}(\lambda) \cdot sensibilite(T_{det}, \lambda) d\lambda \quad (3.1)$$

Pour dresser ce modèle nous avons donc besoin de déterminer  $I_{dark}(T_{det})$  et  $sensibilite(T_{det}, \lambda)$  pour la photo-diode utilisée.

Dans ce but, nous avons utilisé un monochromateur afin de générer un spectre lumineux très étroit dont la longueur d'onde peut être sélectionnée et nous avons mesuré, sur la bande de sensibilité du capteur, le courant généré par l'excitation à cette longueur d'onde. Cette caractérisation a été réalisée en deux temps :

- Une caractérisation précise par pas de 20nm, à 23°C effectuée au *LAAS-CNRS*, selon la Figure 3.15,
- Une caractérisation plus large (pas de 40 à 70nm) aux températures de 10, 30, 50 et 80°C réalisé par *MEAS-France / TE Connectivity*.

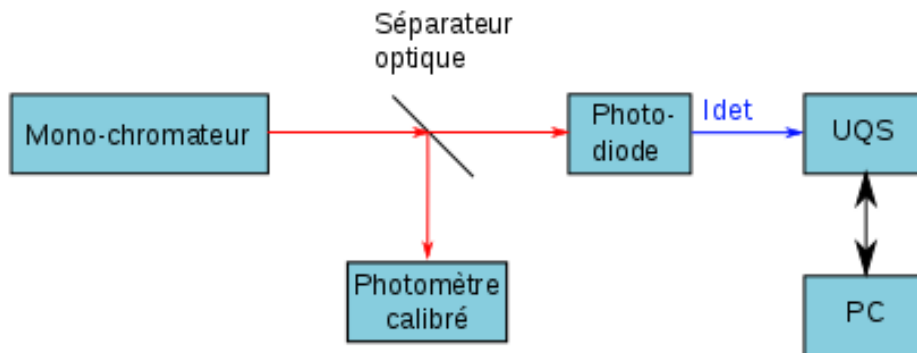


FIGURE 3.15 – Manipulation de caractérisation de la photo-diode au LAAS-CNRS

Le modèle, quant à lui, utilise les tables générées par ces mesures et effectue une interpolation linéaire en température et polynomiale d'ordre 3 en longueur d'onde pour générer le spectre de sensibilité. A partir de là, il calcule le courant de sortie à partir de l'équation 3.1.

### Absorption de l'urée

L'absorption de l'urée correspond à la modification spectrale effectuée par le fluide le long du chemin optique. Cette modification dépend de la température et de la composition du fluide. Dans notre cas, nous ne considérerons que les cas où le fluide est composé d'un mélange d'eau et d'urée, dont les proportions peuvent varier.

Ce modèle peut être représenté selon la Figure 3.16.

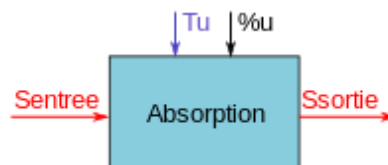


FIGURE 3.16 – Schéma bloc du modèle d'absorption du fluide  
 $S_{sortie}(\lambda, T_u, \%u)$

Pour ce modèle, nous considérons que le spectre de sortie peut être calculé selon l'équation 3.2 pour toute valeur  $\lambda$  de longueur d'onde :

$$S_{sortie}(\lambda, T_u, \%u) = S_{entree}(\lambda) \cdot absorption(\lambda, T_u, \%u) \quad (3.2)$$

où  $absorption(\lambda, T_u, \%u)$  correspond au facteur de transmission de la longueur d'onde  $\lambda$  à la température  $T_u$  pour une concentration de  $\%u$  pourcent d'urée dans l'eau.

Nous avons de plus considéré que le comportement en absorption du fluide était linéaire en fonction de la concentration ce qui donne :

$$absorption(\lambda, T_u, \%u) = absorption_{eau}(\lambda, T_u) + \frac{(32.5 - \%u)}{32.5} absorption_{32.5\%}(\lambda, T_u) \quad (3.3)$$

où nous utilisons dans cette interpolation linéaire l'absorption de l'eau  $absorption_{eau}(\lambda, T_u)$  et l'absorption  $absorption_{32.5\%}(\lambda, T_u)$  d'un fluide composé à 32.5% d'urée à partir des données fournies par *MEAS-France / TE Connectivity* pour ces deux fluides à -10, 0, 10, 20, 30, 40, 50 et 60°C (la Figure 3.4 représente ces fonctions pour une température de 25°C).

Les caractérisations ayant été effectuées par l'entreprise pour un trajet optique de longueur 1cm dans le fluide et le produit ayant un trajet de 3cm dans le fluide, nous avons, de plus, mis à la puissance 3 les niveaux d'absorption fournis pour s'adapter au produit.

Nous obtenons ainsi les courbes d'absorption présentées en Figure 3.9. Ces trois modèles réalisés, nous avons ensuite mis au point un modèle orienté signal, composé des éléments modélisés, du capteur.

### 3.3.4 Modèle orienté signal du système optique

Une fois les éléments constitutifs principaux du capteur modélisés, nous avons choisi de les assembler de façon à dresser un modèle orienté signal du capteur. De cette façon, le modèle global simule en interne le comportement de chaque élément, en propageant les signaux d'un élément à l'autre et en les mettant à jour via les différents éléments traversés.

De par son principe de fonctionnement, nous avons implémenté ce modèle, ainsi que ces sous-modèles, selon un *design pattern publisher subscriber* [12].

Afin d'intégrer différentes altérations du signal, nous avons de plus ajouté un ensemble de gains pouvant être réglés indépendamment :

- Le gain direct, noté  $K_{direct}$  permettant de simuler une opacification du chemin optique (par exemple de la poussière),
- Le gain miroir, noté  $K_{mirror}$  est utilisé pour représenter les variations de puissance d'émission des LEDS, il correspond au facteur de transmission entre la voie miroir et les LEDS,
- Des gains dissociés  $K_{directX}$  ( $X$  correspondant au numéro de chaque LED) et  $K_{mirrorX}$  permettant de simuler une modification de la calibration du capteur pour les premières et une variation individuelle des puissances d'émission pour les secondes.

D'un point de vue implémentation, nous avons de plus choisi d'affecter à la voie directe le gain  $K_{mirror} * K_{direct}$  car une variation de la puissance des LEDS affecte aussi la voie directe.

Ce modèle peut être représenté selon la Figure 3.17.

Nous voyons ainsi que le modèle orienté signal prend en entrées :

- Les courants d'excitation des LEDS comme entrées de commandes,
- Les températures des différents éléments comme grandeurs d'influence,
- La concentration d'urée comme grandeur d'influence,
- Les gains de transmission directs et miroirs comme grandeurs d'influence.

et fournit en sortie :

- les courants mesurés en sortie des photo-diodes.

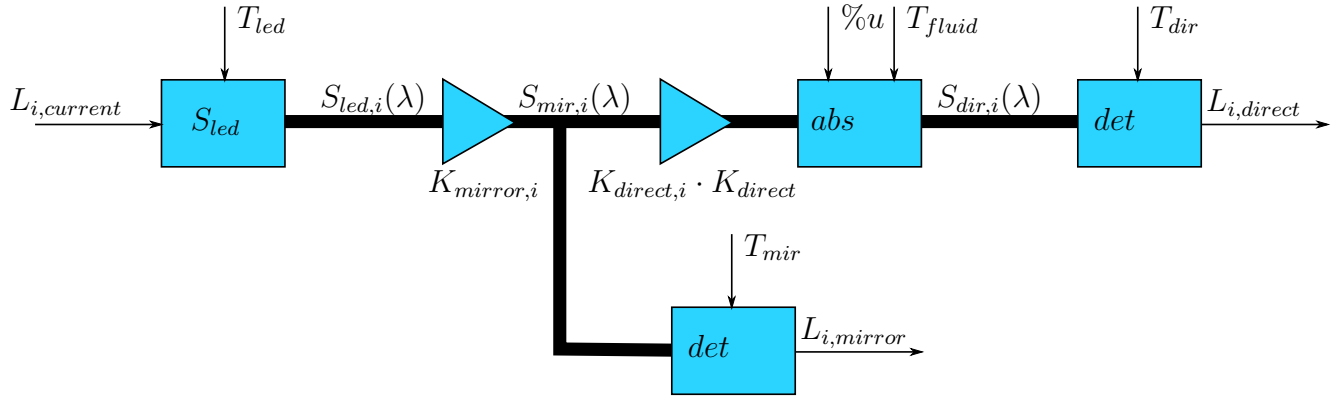


FIGURE 3.17 – Schéma bloc du modèle orienté signal

Les équations du modèle optique sont résumées ici et comportent donc une grande quantité de tables non linéaires :

$$L_{i,mir} = \int_{-\infty}^{+\infty} S_{led,i}(\lambda, T_{led}, L_{i,current}) K_{mir,i} sens(\lambda, T_{mir}) d\lambda \quad (3.4)$$

$$L_{i,dir} = \int_{-\infty}^{+\infty} S_{led,i}(\lambda, T_{led}, L_{i,current}) K_{mir,i} K_{dir} K_{dir,i} abs(\lambda, T_u, \%u) sens(\lambda, T_{dir}) d\lambda \quad (3.5)$$

Ainsi, ce modèle correspond aux contraintes d'interface et de représentation déterminées précédemment.

A ce stade, nous considérons cependant le comportement du système comme statique et instantané. Afin de représenter le comportement dynamique, nous avons donc choisi de séparer le comportement thermique du comportement optique.

### 3.3.5 Une dynamique thermique indépendante

Si on s'attarde sur le comportement physique des composants, on se rend compte que la température de ces derniers a une influence sur leurs sorties. Cependant, cette température n'est influencée que de façon négligeable par le comportement des autres éléments du système. Elle représente de plus un comportement dynamique bien plus lent que la partie optique (considérée comme instantanée aux vues de la période d'échantillonnage de 500ms du système).

Ces différentes considérations nous ont amené à établir le comportement thermique du système comme un sous-système indépendant dont les sorties servent de paramètres d'entrée au modèle optique décrit précédemment. Comme évoqué dans le chapitre précédent 2.5.1, la représentation séparée de ce sous-système linéaire permet de réduire fortement la complexité de l'estimateur global.

Pour dresser ce modèle, nous considérons que le système thermique comprend 2 entrées distinctes :

- La température ambiante, correspondant à la température du fluide dans lequel baigne le capteur,  $T_{fluid}$ ,
- L'auto-échauffement dû à la dissipation par effet Joules  $P_d$  des différents composants (comme présenté ci-dessous, cet échauffement est principalement dû au micro-contrôleur et au driver de bus CAN). Celle-ci est considéré comme constant après allumage.

De plus, il fournit en sorties :

- Les 3 points de mesure de température effectués par les capteurs :  $T_uMes$ ,  $T_{led}Mes$ ,  $T_{dir}Mes$  (respectivement la température du fluide, du point entre les LEDs et la photo-diode miroir et le dernier à côté de la photo-diode directe),
- Les valeurs «réelles» des températures des composants :  $T_{fluid}$ ,  $T_{led}$ ,  $T_{mir}$  et  $T_{dir}$  (respectivement la température du fluide, des LEDs, de la photo-diode miroir et de la photo-diode directe).

Et peut donc se représenter selon la Figure 3.18.

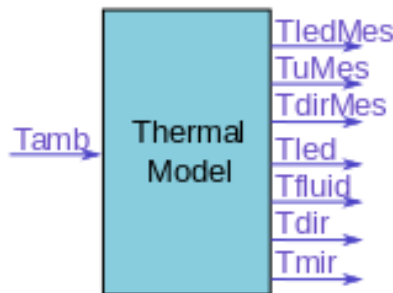


FIGURE 3.18 – Schéma bloc du modèle thermique

Ce comportement thermodynamique étant linéaire par nature, cette modélisation séparée permet de plus de diminuer la complexité des calculs durant la phase d’estimation que nous traiterons ci-dessous.

Pour caractériser et identifier le modèle thermique, deux campagnes de mesures ont été effectuées dans le laboratoire :

- la caractérisation thermique du PCB nu, i.e. sans le boîtier plastique et sans immersion dans le liquide, présentée dans la section 3.3.5 effectuée à l’aide d’une caméra thermique. Cette caractérisation a permis de mettre en évidence les phénomènes d’auto-échauffement et d’orienter le modèle analytique développé dans la section 3.3.5 vers les phénomènes thermiques les plus sensibles ;
- des mesures de réponses à des échelons (présentées dans la section 3.3.5 de températures du fluide) et à des échelons d’auto-échauffement, soit des démarrages à froid. Ces mesures ont permis l’identification paramétrique du modèle.

### Caractérisation de l’auto-échauffement

Les mesures thermiques sur les capteurs UQS ont été répétées à deux occasions au *LAAS-CNRS* : sur la version A du capteur en Décembre 2014 et sur la version B du capteur en Mai 2016.

Pour réaliser ces mesures, nous avons utilisé :

- Une caméra thermique FLIR x6580sc,
- Un PC portable équipé du logiciel de communication CAN *MEAS-France / TE Connectivity* SUS Studio,
- Un boîtier de communication USB/CAN *NI CAN*,
- Une alimentation stabilisée permettant d’alimenter le produit à +5V,
- Un étau permettant de maintenir le produit en position.

Pour réaliser les mesures thermiques sur les capteurs, ces derniers sont montés «PCB nus» : toutes les parties mécaniques ainsi que les guides optiques sont retirés pour qu’il ne reste que le circuit imprimé avec les composants.



Le circuit est alors positionné devant la caméra (Figure 3.19), connecté au boîtier *NI CAN* et à l'alimentation stabilisée (éteinte).

Après mise au point et démarrage de l'enregistrement des données sur la caméra, le produit est alimenté. L'enregistrement se déroule jusqu'à stabilisation des températures sur le produit.

Le test est répété sur les deux faces du circuit imprimé.



FIGURE 3.19 – Installation du capteur devant la caméra IR

**Sur la version A** du produit, nous avons effectué les mesures en deux fois :

- une première mesure sur circuit normal,
- une seconde mesure sur circuit peint en noir (peinture à la bombe, permettant d'homogénéiser l'émissivité thermique).

Les résultats présentés ici correspondent aux résultats de la seconde session de mesures, jugée plus pertinente. Ainsi le produit est positionné tel que vu depuis la caméra (Figure 3.20). Nous obtenons ainsi, en régime stabilisé un profil thermique du produit selon les trois lignes Li1, Li2 et Li3 les températures présentées en Figure 3.21, ainsi que la réponse temporelle présentée en Figure 3.22.

Nous remarquons ainsi que les principaux échauffements sont réalisés par le micro-contrôleur et le driver de bus CAN. De plus, le temps de réponse thermique du système à l'allumage est d'environ 1 minute (donc une stabilisation à 95% de la valeur finale à 3 minutes) et varie légèrement selon les composants concernés.

**Sur la version B**, la procédure précédente a été répétée sur un produit nu et non peint. Le produit a été positionné successivement selon les vues de face (Figure 3.23a), de dos (Figure 3.23b) et de côté (Figure 3.23c) pour effectuer les mesures.

Les résultats obtenus lors de ces mesures, selon les différentes lignes, sont présentés en Figures 3.24 et 3.25.

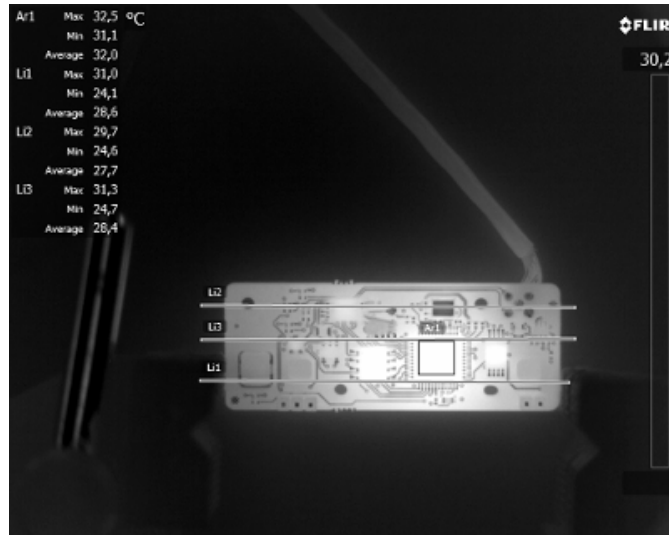


FIGURE 3.20 – Positionnement du capteur vu par la caméra

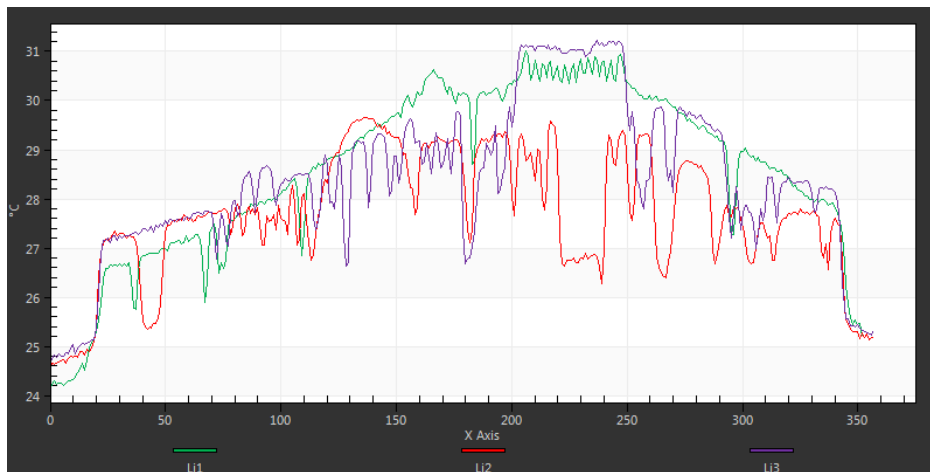


FIGURE 3.21 – Profil thermique du capteur en état stable

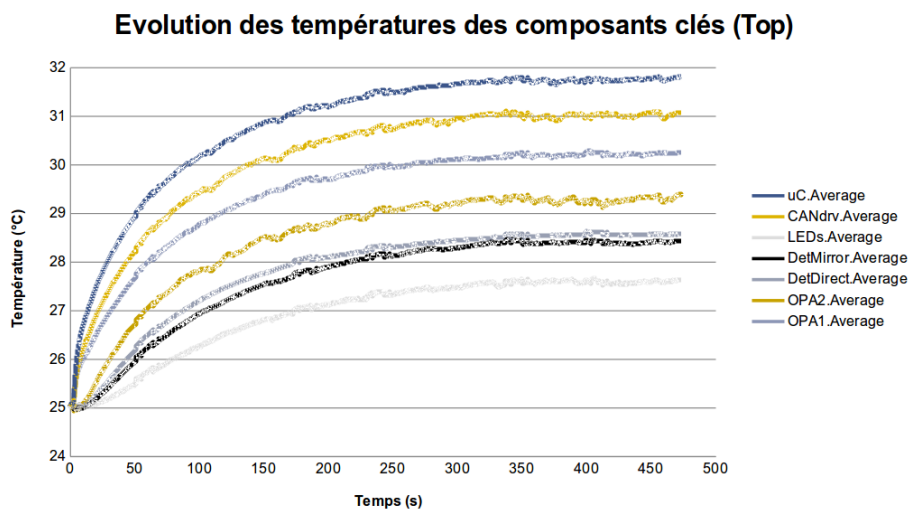


FIGURE 3.22 – Comportement transitoire du capteur



(a) vue de face

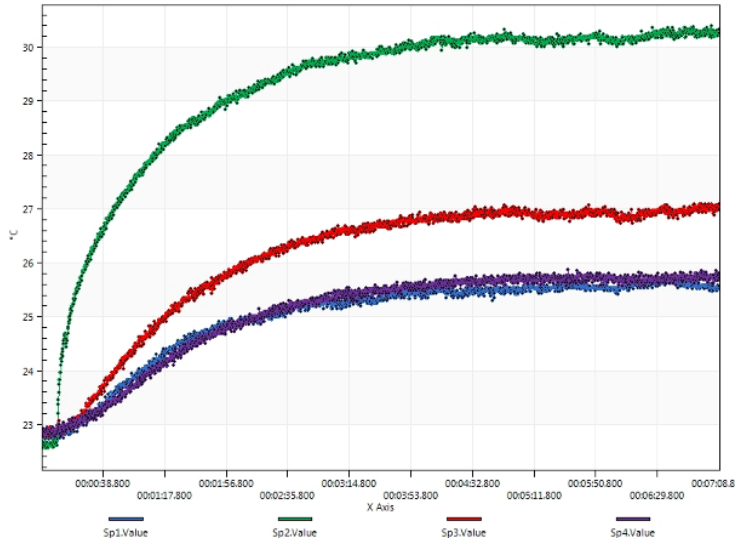


(b) vue de dos

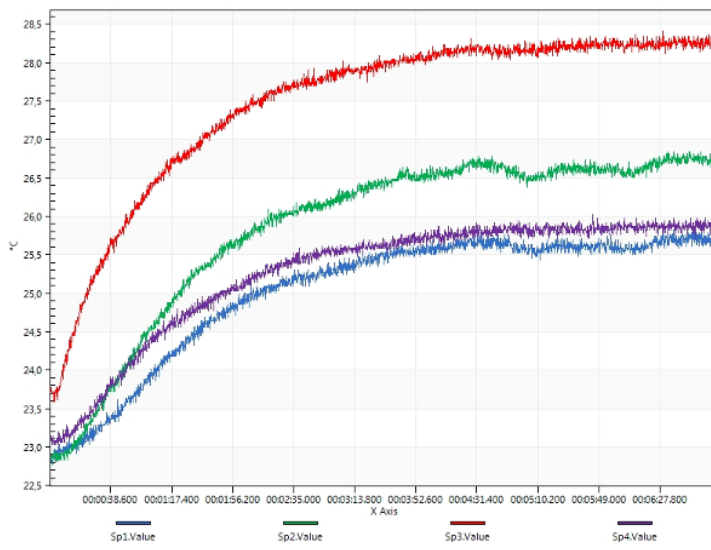


(c) vue de côté

FIGURE 3.23 – Positionnement du capteur vu par la caméra



(a) vue de face



(b) vue de dos

FIGURE 3.24 – Positionnement du capteur vu par la caméra

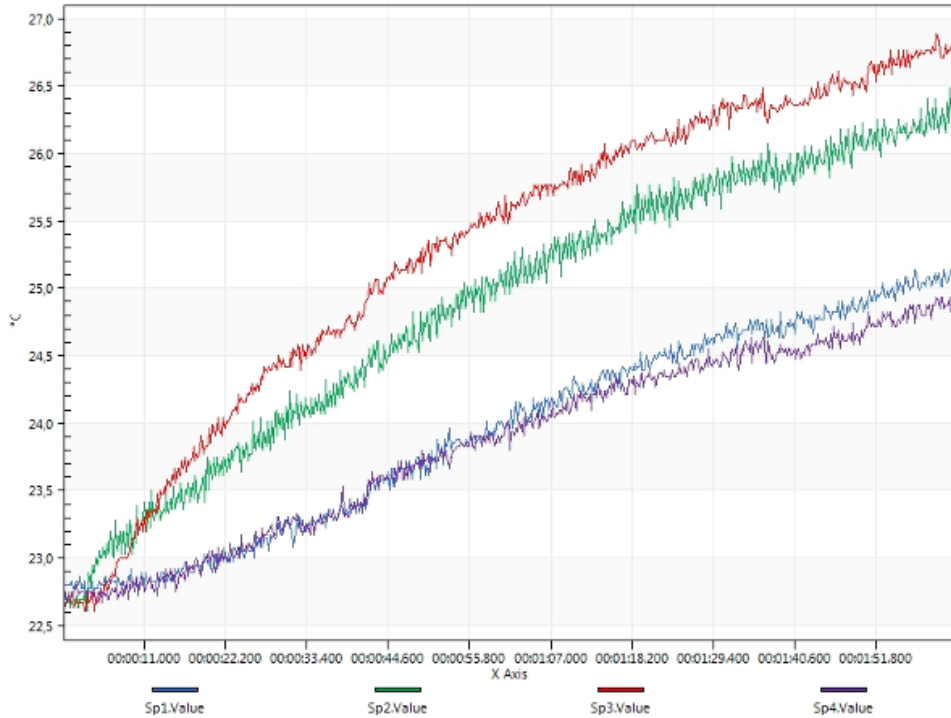


FIGURE 3.25 – Positionnement du capteur vu par la caméra (vue de côté)

Nous pouvons ainsi remarquer que les mesures de côté (Figure 3.25) n’atteignent pas l’état stable, et sont par conséquent à refaire pour être significatives. A part cela, les temps observés sont comparables à ceux mesurés sur la version A du capteur.

**Remarques :** L’émissivité thermique des différents matériaux présent n’étant pas parfaitement connue, les valeurs indiquées par la caméra, ainsi que les différences de températures entre les composants ne sont pas des mesures exactes. La composante la plus intéressante étant donc le comportement dynamique et l’évolution relative des composants.

De plus, le produit étant sorti de son boîtier, cette caractérisation thermique est utilisable uniquement à but indicatif et n’est pas représentative du comportement thermique réel du système en conditions normales d’utilisation. Cette caractérisation permet de mettre en équations les phénomènes thermiques de manière à obtenir un modèle simple, i.e. linéaire et suffisamment précis comme proposé dans la section suivante.

## Modèle thermique

Afin de dresser le modèle thermique du capteur, nous avons considéré la géométrie de ce dernier, puis dressé un modèle électrique équivalent permettant de déterminer les différents échanges thermiques. Pour cela, nous avons utilisé la table d’équivalence présentée en Table 3.1.

Nous avons ainsi dressé le modèle thermique à partir des observations effectuées dans la section 3.3.5 sous la forme du modèle électrique équivalent présenté en Figure 3.26.

Ce modèle considère, d’une part, 4 températures liées à des zones du PCB identifiées par : la température  $T_{cpu}$  dans la zone CPU ; celle de la zone de mesure du trajet direct par  $T_{dir.Mes}$  ; celle de la zone de mesure du trajet miroir et des LEDs d’émission par

Modèle thermique	Modèle électrique
conduction thermique	conductance électrique
capacité thermique	capacité électrique
flux thermique	courant électrique
température	tension

TABLE 3.1 – Correspondance entre les composants thermique et les composants électriques équivalents

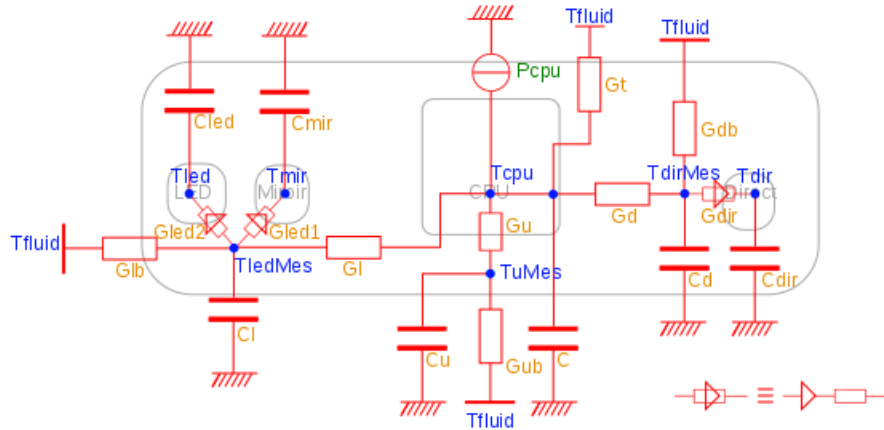


FIGURE 3.26 – Modèle électrique équivalent au système thermique

$T_{LEDMes}$ ; et celle de la zone de mesure de la température d'urée par  $T_uMes$ . On modélise le flux thermique venant de l'échauffement de la CPU et du transmetteur CAN par une source de courant  $P_{cpu}$  affectant directement  $T_{cpu}$ .

Nous modélisons par les conductances  $G_l$ ,  $G_u$ , et  $G_d$  l'échange thermique entre la zone CPU et les zones voisines. Les échanges thermiques entre chaque zone et le fluide via le boîtier sont modélisés par  $G_{lb}$ ,  $G_{ub}$ ,  $G_{db}$  et  $G_t$ . Ainsi le modèle est totalement couplé de sorte que chacune des températures peut influencer l'autre par le biais du phénomène de conduction thermique.

Les observations sur les réponses en échelon thermique et sur la caméra thermique ont permis de mettre en évidence le fait que les capteurs de température situés dans chacune de ces zones évoluent, du fait de leur emplacement sur le PCB, avant que les éléments actifs de la chaîne optique ne soient réellement affectés par ces variations thermiques. Afin de simplifier le modèle et son identification paramétrique nous avons représenté ce phénomène de retard par une dynamique de premier ordre. Ainsi la température  $T_{ledMes}$  correspondant au capteur de température est retardée par un premier ordre via  $G_{led1}$  et  $C_{mir}$  pour donner une température  $T_{mir}$  affectant le détecteur de retour miroir. Ne souhaitant pas modéliser un couplage thermique mais un phénomène de retard causal de  $T_{ledMes}$  vers  $T_{mir}$ , nous considérons qu'une adaptation d'impédance est effectuée par un ampli de gain unitaire (symbole formé d'un triangle sur le schéma 3.26) commandant le premier ordre. Il en est de même pour  $T_{led}$ , donnée elle aussi par un premier ordre ( $G_{led2}$  et  $C_{led}$ ) issue de  $T_{ledMes}$  et  $T_{dir}$  issue de  $T_{dirMes}$  via  $G_{dir}$  et  $C_{dir}$ .

A partir de là, nous pouvons déterminer les équations différentielles nous permettant de déterminer le comportement des différents points de température en fonction de valeurs des divers paramètres (valeurs des composants). Par exemple, l'équation 3.6 représente

$T_uMes$ .

Nous considérons les valeurs du vecteur d'état  $\mathbf{X} = \begin{pmatrix} T_{cpu} \\ T_uMes \\ T_{led}Mes \\ T_{dir}Mes \\ T_{led} \\ T_{dir} \\ T_{mir} \end{pmatrix}$  composé de 7 températures jugées pertinentes pour notre modèle.

$$\frac{dT_uMes}{dt}C_u = (T_{cpu} - T_uMes)G_u + (T_{fluid} - T_uMes)G_{ub} \quad (3.6)$$

En dressant l'ensemble des équations différentielles des températures composant le vecteur d'état, nous pouvons déterminer les matrices d'évolution  $A$  (3.7) et de commande  $B$  (3.8).

$$[\mathbf{A}] = \begin{bmatrix} \frac{-(G_u+G_l+G_d+G_t)}{C_c} & \frac{G_u}{C_c} & \frac{G_l}{C_c} & \frac{G_d}{C_c} & 0 & 0 & 0 \\ \frac{G_u}{C_u} & \frac{-(G_u+G_{ub})}{C_u} & 0 & 0 & 0 & 0 & 0 \\ \frac{G_l}{C_l} & 0 & \frac{-(G_l+G_{lb})}{C_l} & 0 & 0 & 0 & 0 \\ \frac{G_d}{C_d} & 0 & 0 & \frac{-(G_d+G_{db})}{C_d} & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{\tau_{led}} = \frac{G_{led2}}{C_{led}} & 0 & -\frac{1}{\tau_{led}} & 0 & 0 \\ 0 & 0 & \frac{1}{\tau_{dir}} = \frac{G_{dir}}{C_{dir}} & 0 & 0 & -\frac{1}{\tau_{dir}} & 0 \\ 0 & 0 & 0 & \frac{1}{\tau_{mir}} = \frac{G_{led1}}{C_{mir}} & 0 & 0 & -\frac{1}{\tau_{mir}} \end{bmatrix} \quad (3.7)$$

$$[\mathbf{B}] = \begin{bmatrix} \frac{1}{C_c} & \frac{G_t}{C_c} \\ 0 & \frac{G_{ub}}{C_u} \\ 0 & \frac{G_{lb}}{C_l} \\ 0 & \frac{G_{db}}{C_d} \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \quad (3.8)$$

Ainsi que les matrices de mesure  $[\mathbf{C}]$  (3.9) et  $[\mathbf{D}]$  (3.10) permettant d'obtenir le vecteur

$$\text{de mesures } \mathbf{Y} = \begin{pmatrix} T_uMes \\ T_{led}Mes \\ T_{dir}Mes \\ T_{fluid} \\ T_{led} \\ T_{mir} \\ T_{det} \end{pmatrix}.$$

$$[\mathbf{C}] = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \quad (3.9)$$

$$[\mathbf{D}] = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \quad (3.10)$$

Nous remarquons que le vecteur de sortie comprend les véritables mesures issues de capteurs, i.e. toutes les variables se terminant par *Mes*, et des températures n'étant pas directement mesurées mais qui sont estimées par l'observateur d'état que l'on construit à partir de ce modèle dans la section 3.3.7. Ces dernières sont donc des mesures issues du capteur logiciel.

Ainsi, l'équation d'évolution du système s'obtient sous la forme discrete 3.11 en considérant l'équation comme un système échantillonné–bloqué vu dans la section 1.2.4. La fréquence d'échantillonnage  $T_e$ , d'une seconde, étant rapide devant les constantes de temps observées, de l'ordre de la minute, la forme simplifiée de l'expression discrète est choisie. L'équation de mesure, augmentée des températures virtuellement ajoutées par le capteur logiciel, reste inchangée sous la forme de l'équation 3.12.

$$\mathbf{X}_{k+1} = e^{[\mathbf{A}]T_e} \mathbf{X}_k + T_e [\mathbf{B}] \mathbf{U}_k \quad (3.11)$$

$$\mathbf{Y}_k = [\mathbf{C}] \mathbf{X}_k + [\mathbf{D}] \mathbf{U}_k \quad (3.12)$$

avec  $T_e$  la période d'échantillonnage du système.

Afin d'identifier les paramètres de ce modèle, une campagne de mesure (présentée dans la suite) effectue des échelons de température de fluide et d'auto-échauffement permettant d'identifier les valeurs numériques des paramètres.

### Procédure d'étalonnage et caractérisation

Une fois la structure du modèle dressée, il faut attribuer des valeurs aux différents paramètres de conductances et de capacités. Pour cela, nous avons effectué sur un produit UQS les essais suivants :

- Un démarrage à froid : le produit est positionné non alimenté dans un bain d'eau à température ambiante durant 10min afin d'être parfaitement thermalisé. Le produit est ensuite allumé, toujours dans le bain, et ses mesures sont enregistrées sur une durée de 20min dans un fichier par le biais d'une liaison CAN vers un PC.



- Un échelon thermique : le produit est placé dans un bain d'eau à température ambiante, alimenté jusqu'à stabilisation des températures (ou à la suite du test précédent). A l'instant  $T_1$ , nous ajoutons une grande quantité d'eau à ébullition (environ 2 fois le volume du bain initial) et nous mélangeons délicatement pour évacuer les bulles du trajet optique et homogénéiser les températures du fluides. Durant tout le test, ainsi que les 20 minutes suivantes, les mesures sont sauvegardées dans un fichier sur le PC de test.

À partir des données produites par ces deux tests, nous sommes en mesure d'identifier les constantes de temps des impacts de chaque entrée sur les températures de sortie (traduites par des rapports conductance/capacité), ainsi que les rapports à appliquer sur les impacts (traduits par des rapports de conductances).

L'identification de ces paramètres est relativement aisée pour les fonctions de transfert vers les températures mesurées  $T_uMes$ ,  $T_{led}Mes$  et  $T_{dir}Mes$ . Elle est cependant plus difficile en ce qui concerne les températures réelles des composants  $T_{led}$ ,  $T_{mir}$  et  $T_{dir}$  car ces informations ne sont pas directement observables au travers des mesures, mais seulement par extrapolation au niveau des mesures optiques (variations des puissances émises par les LEDs, des courants d'obscurité des photo-diodes et de leurs sensibilités).

La résolution analytique des paramètres du modèle d'évolution discret  $e^{\mathbf{A}T_e}$  étant irréalisable, nous choisissons de résoudre les paramètres d'évolution dans le domaine continu. Pour cela, nous commençons par résoudre les paramètres  $G_u, G_l, G_d, G_t, C_c, C_u, C_l$  et  $C_d$  en partant d'une valeur initiale, et en cherchant à minimiser l'erreur de l'expression :

$$\sum_k (\mathbf{Y}_{mes,k} - \tilde{\mathbf{Y}}_k(\Gamma)) \cdot (\mathbf{Y}_{mes,k} - \tilde{\mathbf{Y}}_k(\Gamma))^T \quad (3.13)$$

en utilisant une fonction de recherche itérative de minimum [13], avec :

- $\Gamma = (G_u, G_l, G_d, G_t, C_c, C_u, C_l, C_d)^T$  le jeu de paramètres à estimer,
- $\mathbf{Y} = (T_{uMes}, T_{ledMes}, T_{dirMes})^T$  les mesures effectuées lors de la phase de calibration,
- $\tilde{\mathbf{Y}}(\Gamma) = (\tilde{T}_{uMes}, \tilde{T}_{ledMes}, \tilde{T}_{dirMes})^T$  les mesures estimées d'après le modèle thermique paramétré par  $\Gamma$ .

En appliquant cette technique sur les mesures de calibration thermique, nous avons pu déterminer le jeu de paramètres présentés en Table 3.2.

Paramètre	Valeur
$G_t$	1.88789722e-08
$C_c$	9.99355658e-03
$G_u$	3.90177558e-07
$G_{ub}$	6.80321536e-08
$C_u$	1.44679623e-05
$G_d$	3.79327528e-05
$G_{db}$	1.59315523e-05
$C_d$	1.61689630e-03
$G_l$	7.82413025e-04
$G_{lb}$	3.61409392e-04
$C_l$	8.89679715e-03

TABLE 3.2 – Valeurs des paramètres de calibration du modèle thermique

A partir de là, les paramètres  $\tau_{led}$ ,  $\tau_{mir}$  et  $\tau_{dir}$  ont été déterminés manuellement par dichotomie afin de minimiser l'erreur de mesure au niveau des courants mesurés ( $L2_{direct}$ ,  $L3_{direct}$ ,  $L3_{direct}$ ,  $L2_{mirror}$ ,  $L3_{mirror}$ ,  $L4_{mirror}$ ) sur le modèle optique alimenté par les mesure logicielles  $T_{led}$ ,  $T_{dir}$  et  $T_{mir}$  de l'estimateur thermique. Nous obtenons ainsi les résultats présentés dans la Table 3.3.

Paramètre	Valeur
$\tau_{led}$	100
$\tau_{dir}$	50
$\tau_{mir}$	100

TABLE 3.3 – Valeurs des paramètres de calibration manuelle

Une fois le modèle calibré, nous sommes en mesure de le tester en comparant son comportement aux mesures réelles, soit lors d'un démarrage à froid dans un fluide de température connue, soit par rapport à des variations de température de profil connu réalisé en cryostat, ou en confrontant l'estimation aux mesures pour l'échelon thermique (Figure 3.27).

### 3.3.6 Modèle complet

Une fois ces deux éléments mis en place, nous pouvons composer le modèle orienté signal complet du système à partir du modèle thermique et du modèle optique. Ce modèle peut être représenté selon le schéma bloc 3.28, et est détaillé en Figure 3.29.

### 3.3.7 Estimation et stratégies de calibration

Nous disposons maintenant d'un système composé de deux sous-modèles :

- le modèle thermique, linéaire, indépendant de la partie optique et pré-calibré,
- le modèle optique orienté signal, commandé par le modèle thermique et pouvant faire l'objet d'une calibration pour fixer les gains de transmission  $K_{direct}$ ,  $K_{directX}$ ,  $K_{mirror}$  et  $K_{mirrorX}$  ainsi que la concentration d'urée.

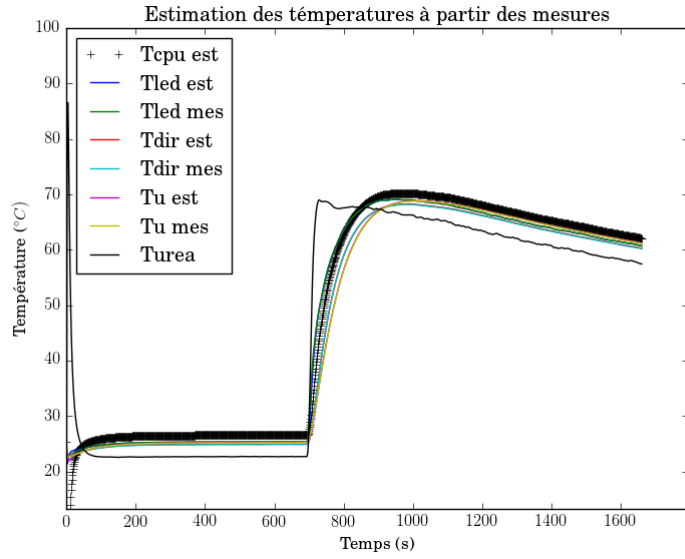
En s'appuyant sur ces modèles nous pouvons donc dresser un ensemble d'estimateurs de Kalman permettant de répondre à la problématique d'estimation des paramètres permettant de fournir une concentration d'urée fiable en sortie.

#### Estimateur thermique

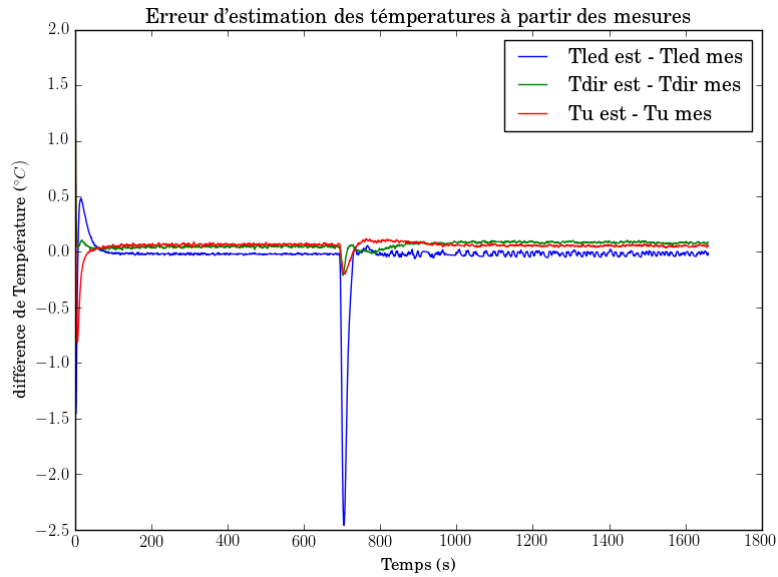
Dans un premier temps, le modèle thermique étant parfaitement établi et linéaire, nous pouvons mettre en place un estimateur nous permettant de déterminer les températures des différents éléments du système ainsi que du fluide à partir des mesures effectuées par les capteurs.

Pour cela, nous modifions le modèle précédent tel que :

- La température du fluide, jusqu'à présent utilisée comme entrée du système, devienne un état interne du système (qui correspond aux données que nous voulons estimer),
- L'entrée de commande de la puissance de dissipée  $P_{cpu}$  est une constante,
- Les sorties du système sont les mesures effectuées par le capteur  $T_uMes$ ,  $T_{led}Mes$  et  $T_{dir}Mes$ .



(a) Estimation des températures



(b) Erreurs d'estimation

FIGURE 3.27 – Comparaison du modèle Thermique avec les mesures réelles lors d'un échelon thermique

Nous obtenons ainsi pour vecteurs de commande, d'état (3.14) et de mesure (3.15) :

$$\mathbf{X} = \begin{pmatrix} T_{cpu} \\ T_u Mes \\ T_{led} Mes \\ T_{dir} Mes \\ T_{fluid} \\ T_{led} \\ T_{dir} \\ T_{mir} \end{pmatrix} \quad (3.14)$$

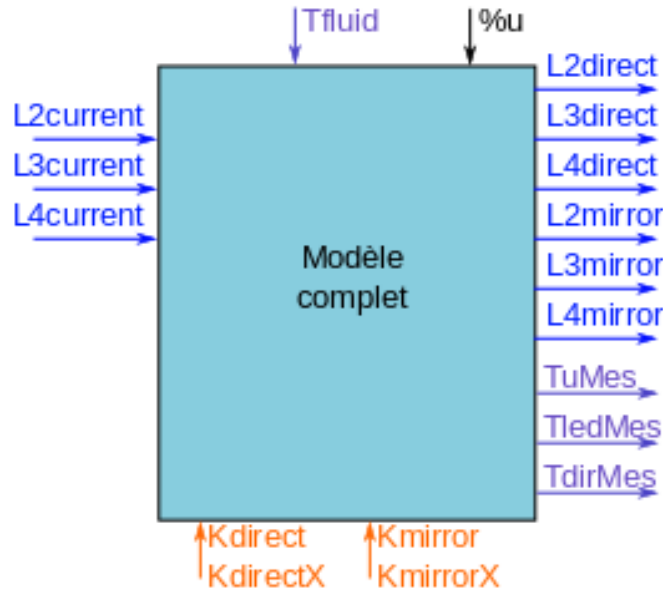


FIGURE 3.28 – Modèle complet du capteur

$$\mathbf{Y} = \begin{pmatrix} T_u Mes \\ T_{led} Mes \\ T_{dir} Mes \end{pmatrix} \quad (3.15)$$

De plus, le vecteur de commande devient (3.16) :

$$\mathbf{U} = (P_{cpu}) \quad (3.16)$$

Nous pouvons ainsi déduire les matrices d'évolution (3.17, 3.18) et de mesure (3.19, 3.20) :

$$[\mathbf{A}] = \begin{bmatrix} \frac{-(G_u+G_l+G_d+G_t)}{C_c} & \frac{G_u}{C_c} & \frac{G_l}{C_c} & \frac{G_d}{C_c} & \frac{G_t}{C_c} & 0 & 0 & 0 \\ \frac{G_u}{C_u} & \frac{-(G_u+G_{ub})}{C_u} & 0 & 0 & \frac{G_{ub}}{C_u} & 0 & 0 & 0 \\ \frac{G_l}{C_l} & 0 & \frac{-(G_l+G_{lb})}{C_l} & 0 & \frac{G_{lb}}{C_l} & 0 & 0 & 0 \\ \frac{G_d}{C_d} & 0 & 0 & \frac{-(G_d+G_{db})}{C_d} & \frac{G_{db}}{C_d} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{\tau_{led}} = \frac{G_{led2}}{C_{led}} & 0 & -\frac{1}{\tau_{led}} & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{\tau_{dir}} = \frac{G_{dir}}{C_{dir}} & 0 & 0 & -\frac{1}{\tau_{dir}} & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{\tau_{mir}} = \frac{G_{led1}}{C_{mir}} & 0 & 0 & -\frac{1}{\tau_{mir}} \end{bmatrix} \quad (3.17)$$

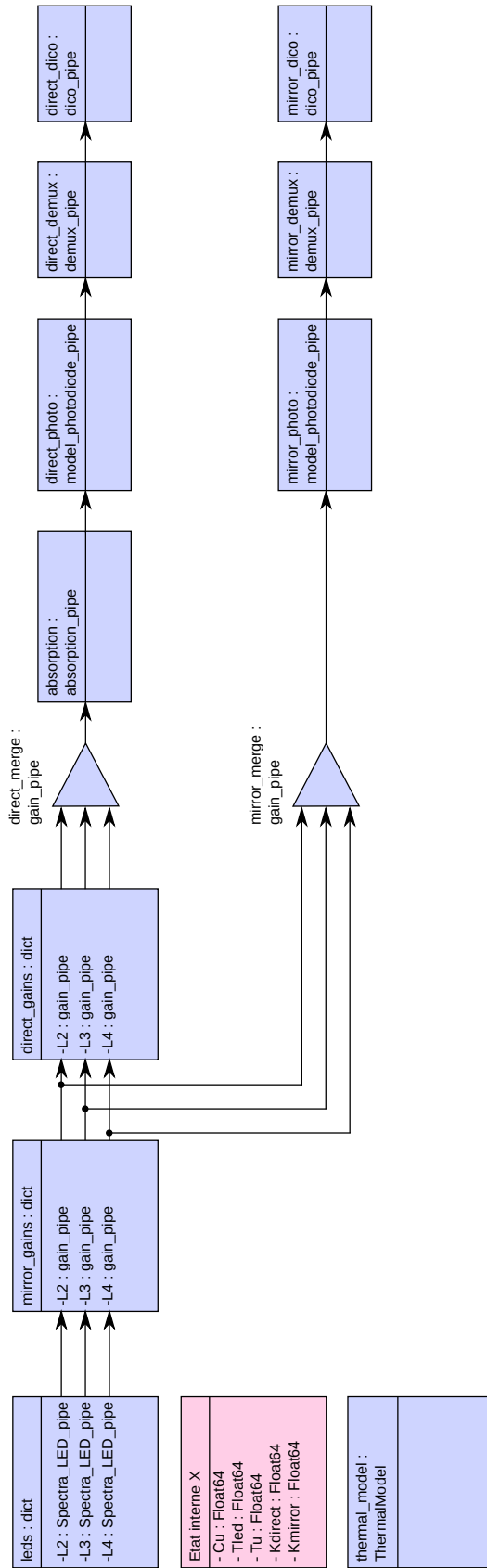


FIGURE 3.29 – Modèle orienté signal complet

$$[\mathbf{B}] = \begin{bmatrix} \frac{1}{C_c} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (3.18)$$

$$[\mathbf{C}] = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (3.19)$$

$$[\mathbf{D}] = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad (3.20)$$

A partir de ces équations, nous sommes en mesure de mettre en place l'estimateur de Kalman linéaire permettant d'obtenir les températures des différents composants, qui nous serviront de commande pour les estimations sur les modèles orientés signaux.

### En mode calibration

Lors de la phase de calibration, le capteur est plongé dans un bain dont la concentration d'urée est connue, les paramètres que nous souhaitons estimer étant les gains directs  $Kdirect$ ,  $KdirectX$  ainsi que les gains miroir  $Kmirror$  et  $KmirrorX$ .

Le vecteur de commande du modèle orienté signal est donc  $\mathbf{U}_{calib} = \begin{pmatrix} L2_{current} \\ L3_{current} \\ L4_{current} \\ u\% \\ T_{led} \\ T_{dir} \\ T_{fluid} \\ T_{mir} \end{pmatrix}$ .

Le vecteur d'état correspondant est  $\mathbf{X}_{calib} = \begin{pmatrix} Kdirect2 \\ Kdirect3 \\ Kdirect4 \\ Kmirror2 \\ Kmirror3 \\ Kmirror4 \end{pmatrix}$ .

Afin de limiter la redondance, et donc les colinéarités dans le modèle, nous avons choisi de combiner les gains communs  $Kdirect$  et  $Kmirror$  avec les gains  $Kdirect4$  et  $Kmirror4$ , vu que la LED4 n'est impactée que de façon négligeable par la concentration d'urée.

Les sorties du modèle quant à elles sont les courants de sortie :  $\mathbf{Y}_{calib} = \begin{pmatrix} L2_{direct} \\ L2_{mirror} \\ L3_{direct} \\ L3_{mirror} \\ L4_{direct} \\ L4_{mirror} \end{pmatrix}$

A partir de ces données, nous avons utilisé un modèle d'évolution identitaire (la matrice d'évolution  $A$  est une matrice identité de la taille du vecteur d'état), et nous avons utilisé un filtre  $HUKF$  afin de traiter l'estimation du vecteur d'état.

Lorsque l'estimation a convergé, les paramètres  $K_{direct2}$ ,  $K_{direct3}$  et  $K_{direct4}$  contiennent la calibration du capteur et peuvent être sauvegardés pour être de nouveau utilisés lors de la phase d'estimation de la concentration.

### En mode estimation de concentration

Une fois le capteur calibré, les gains directs de calibration sont figés et utilisés comme commandes de l'estimateur de concentration. Dans cette phase, la concentration d'urée n'est plus connue et doit donc être estimée.

Ainsi, le vecteur de commande du modèle devient  $\mathbf{U}_{estim} = \begin{pmatrix} L2_{current} \\ L3_{current} \\ L4_{current} \\ T_{led} \\ T_{dir} \\ T_{fluid} \\ T_{mir} \\ K_{direct2} \\ K_{direct3} \\ K_{direct4} \end{pmatrix}$ .

Le vecteur d'état est diminué des gains directs indépendants. On y ajoute cependant le gain direct commun  $K_{direct}$  ainsi que la concentration d'urée  $u\%$ . Ainsi, il devient

$$\mathbf{X}_{estim} = \begin{pmatrix} u\% \\ K_{mirror2} \\ K_{mirror3} \\ K_{mirror4} \\ K_{direct} \end{pmatrix}.$$

Nous gardons les gains miroirs indépendants  $K_{mirrorX}$  afin de prendre en compte les variations de puissance d'émission des LEDs. Le gain direct commun  $K_{direct}$  quant à lui correspond à une opacification homogène du chemin optique (vieillessement des plastiques, poussière...).

Le vecteur de mesure quant à lui reste identique à celui utilisé lors de la phase de calibration  $\mathbf{Y}_{estim} = \mathbf{Y}_{calib}$ .

### 3.3.8 Simulation

Une fois l'estimateur et le modèle mis en place, nous sommes en mesure d'effectuer les premières simulations sur modèle afin de valider la cohérence globale de la démarche.

Pour cela, nous avons réalisé trois types de simulations que nous allons décrire ici :

- une simulation de calibration et de convergence en estimation, avec des variations de température et de concentration,
- une simulation de calibration et de convergence en estimation avec des variations de l'opacité du chemin optique,
- une simulation de calibration et de convergence en estimation avec des variations des puissances d'émission des LEDs.

#### Procédure de calibration du capteur

Le profil de calibration utilisé est identique pour les trois tests et est décrit comme suit (Figure 3.30) :

- La concentration d'urée est connue et de 32.5%,
- Les gains directs sur les LEDs 2, 3 et 4 sont respectivement 0.8, 1.1 et 1.3,
- Le gain global (commun à toutes les LEDs) sur la voie miroir est de 0.9,
- Tous les autres gains sont à 1.0,
- la procédure de calibration est effectuée sur une durée de 3000 secondes (50 min),
- les 100 premières secondes, la température du fluide est à 5°C,
- ensuite, la température monte selon une rampe entre 5°C et 60°C durant les 2400 secondes (40 minutes) suivantes,
- sur les 500 dernières secondes, la température reste constante à 60°C.

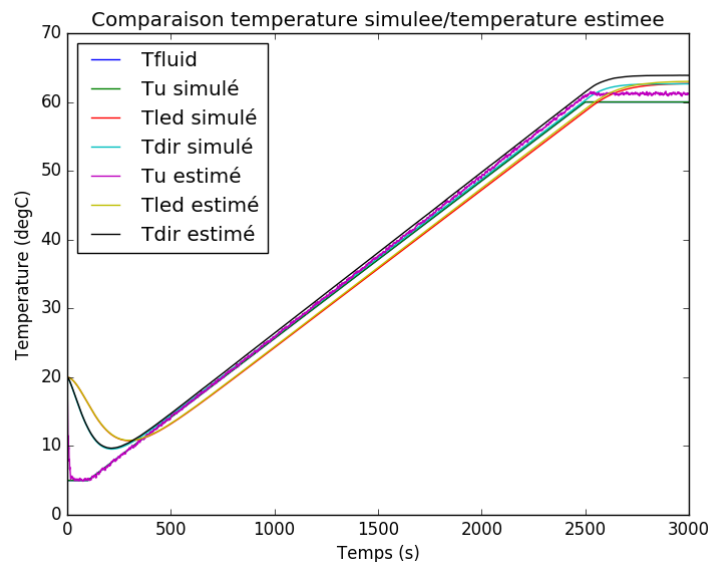
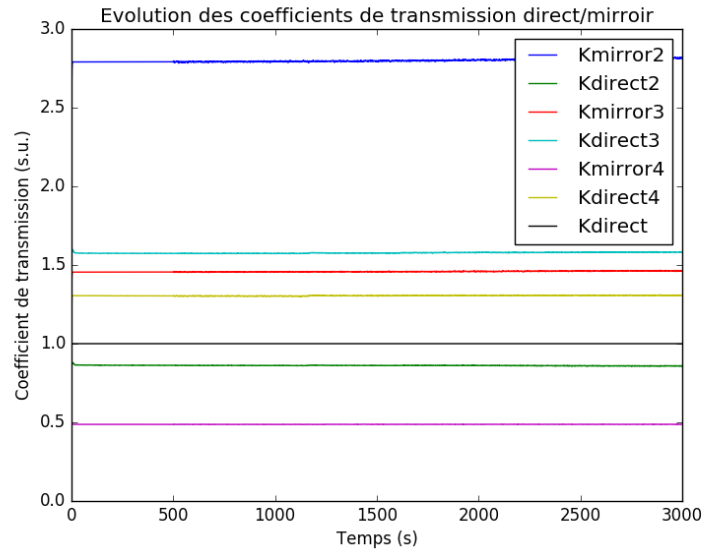


FIGURE 3.30 – Profil thermique de calibration

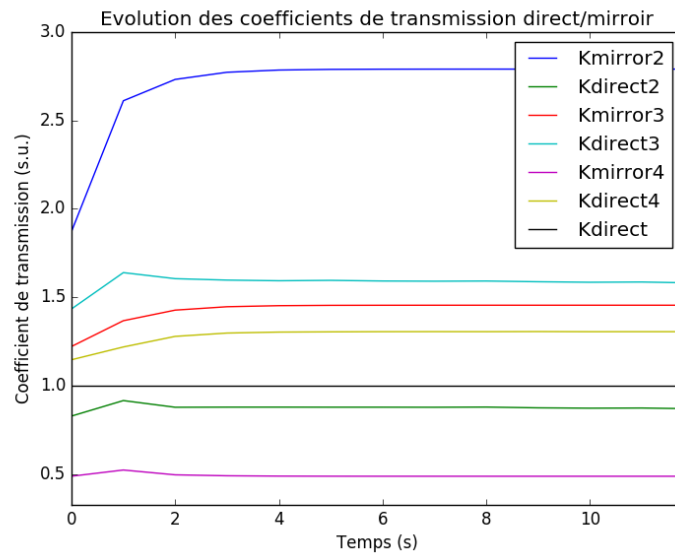
Nous constatons que le modèle thermique (Figure 3.30) et le modèle de calibration optique (Figure 3.31) convergent rapidement vers les valeurs paramétrées pour la simulation.

Une fois la calibration effectuée, les valeurs de gains sont sauvegardées comme paramètres pour l'estimateur d'urée.





(a) Scenario complet



(b) Début de scénario

FIGURE 3.31 – Evolution des gains estimés lors de la phase de calibration

### Convergence lors d'une variation de température

La première étape de validation consiste ensuite à valider le fonctionnement de l'estimateur lors de variations de température et de concentration d'urée. Pour cela, nous avons mis en place le scénario suivant :

- Durant les 800 premières secondes, la température est de  $60^{\circ}C$  et la concentration de 30% d'urée,
- Entre 800 et 2200 secondes, la concentration diminue de façon linéaire de 30% à 10%,
- Puis les paramètres sont stables de 2200 à 3000s,
- Finalement la température diminue de  $60^{\circ}C$  à  $10^{\circ}C$  entre 3000 et 4000s, avant de finir sur un état stable jusqu'à 6000s.

Durant tout le test, les bruits de mesures gaussiens d'écart-types :

- $0.2^{\circ}C$  pour les températures,
- 5 counts pour les mesures optiques.

Les résultats de simulation de la température sont présentés en Figure 3.32 et 3.33.

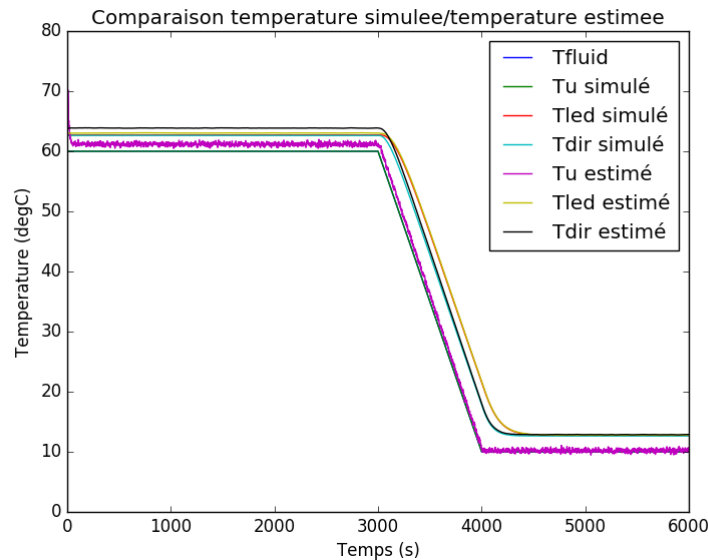


FIGURE 3.32 – Comparaison de l'estimation et de la simulation de température (variation de température et de concentration)

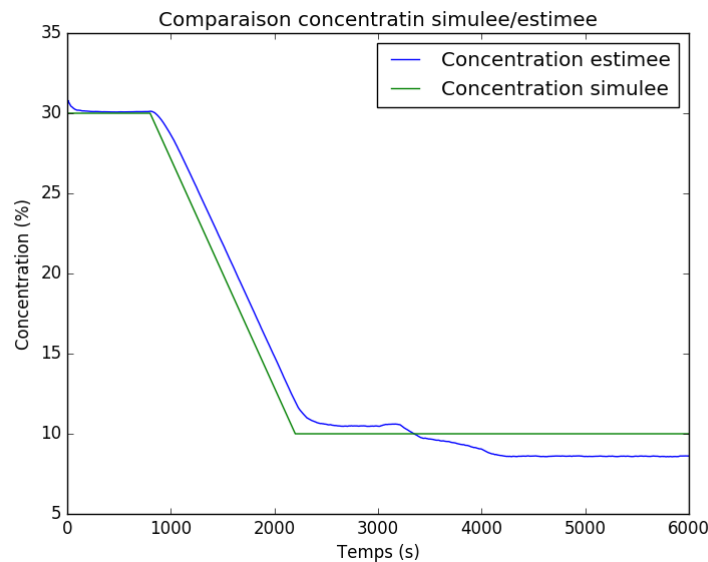


FIGURE 3.33 – Comparaison de l'estimation et de la simulation de concentration (variation de température et de concentration)

Nous obtenons ainsi une erreur moyenne et un écart-type pour chaque paramètre tels que présentés dans le tableau 3.4.

Avec une erreur maximale d'estimation de l'ordre de 2% d'urée sur la plage de test, nous considérons que l'estimateur mis en place est suffisamment fiable pour effectuer des simulations plus avancées.

Paramètre	Erreur Moyenne ( $^{\circ}C$ )	Ecart-type ( $^{\circ}C$ )
$T_{fluid}$ ( $^{\circ}C$ )	-0.892	0.629
$T_{led}$ ( $^{\circ}C$ )	-0.218	0.122
$T_{direct}$ ( $^{\circ}C$ )	-0.812	0.469
$u\%$ (%)	-0.0224	1.224

TABLE 3.4 – Erreurs sur l'estimation des paramètres principaux

### Convergence lors d'une variation d'opacité sur le chemin optique

Pour la seconde étape de validation, nous cherchons à estimer l'impact d'une variation de gain direct (correspondant à l'opacité du fluide le long du chemin optique, par variation de la poussière par exemple). Pour cela, nous avons mis en place le scénario de simulation suivant :

- Lors du test, la température est fixée à  $40^{\circ}C$  et la concentration d'urée à 30%,
- Les 500 premières secondes, le gain direct est fixé à 1.3, stable,
- entre 500 et 3500 secondes, le gain direct diminue progressivement de 1.3 à 0.7, de façon linéaire,
- entre 3500 et 6000 secondes, le gain est maintenu stable à 0.7.

Les bruits de mesure sont maintenus aux niveaux du test précédent.

Nous obtenons ainsi l'estimation de la concentration d'urée présentée en Figure 3.34.

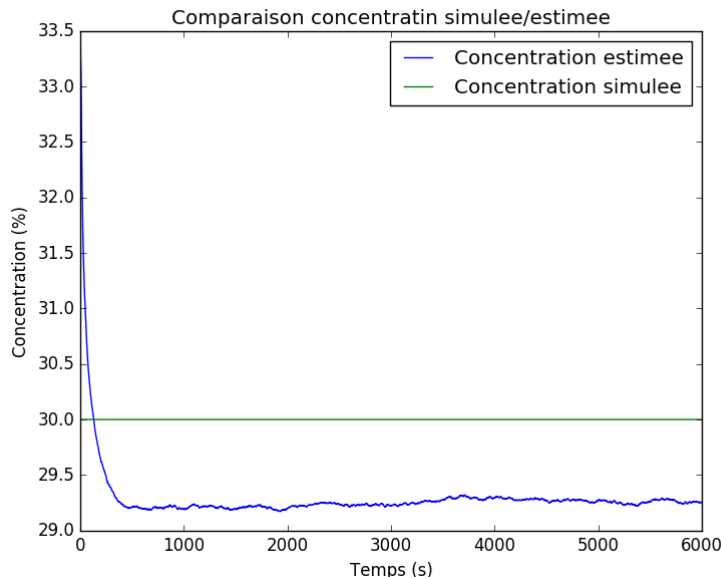


FIGURE 3.34 – Comparaison de l'estimation et de la simulation de concentration (variation de gain direct)

Nous remarquons que la variation de gain direct utilisée lors de la simulation n'a pas d'impact sur l'estimation de la concentration d'urée, avec une erreur moyenne d'estimation de 0.706% et un écart-type de 0.297% sur la durée du test.

Ainsi, la stratégie mise en place pour compenser les variations communes de gain direct (et donc, les obstructions telles que la poussière) présente des résultats prometteurs.

## Convergence lors d'une variation de puissance des LEDs

Pour le test final, nous voulons vérifier l'impact d'une variation de puissance d'émission des LEDs sur l'estimation de la concentration d'urée. Cette variation étant simulée par une variation du gain miroir correspondant à la LED incriminée, et pouvant retranscrire un vieillissement du composant, une dégradation de celui-ci ou une variation de la commande.

Pour ce test, nous avons simulé le scénario suivant :

- Lors du test, la température est fixée à  $40^{\circ}\text{C}$  et la concentration d'urée à 30%,
- Les 100 premières secondes, le gain miroir global est fixé à 0.9, stable et les gains miroir individuels sont fixés à 1.0,
- entre 100 et 2000 secondes, le gain miroir de la LED2 diminue progressivement de 1.0 à 0.9, de façon linéaire,
- entre 2000 et 3500 secondes, le gain miroir de la LED4 diminue de 1.0 à 0.8,
- entre 3500 et 6000 secondes, les gains sont maintenus stables.

Nous obtenons ainsi l'estimation de la concentration d'urée présentée en Figure 3.35.

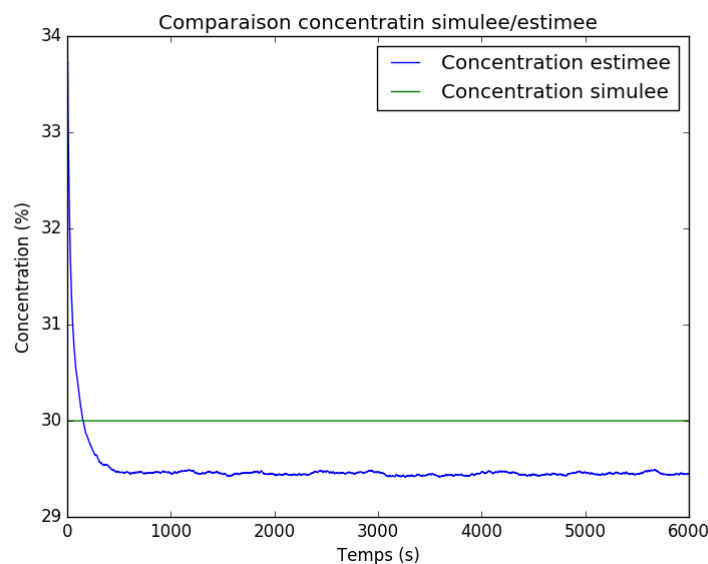


FIGURE 3.35 – Comparaison de l'estimation et de la simulation de concentration (variation de gains miroirs)

Nous remarquons que la variation de gain miroir utilisée lors de la simulation n'a pas d'impact sur l'estimation de la concentration d'urée, avec une erreur moyenne d'estimation de 0.499% et un écart-type de 0.294% sur la durée du test.

Ainsi, la stratégie mise en place pour compenser les variations individuelles de gain miroir (et donc, les variations de puissance d'émission des LEDs) présente des résultats prometteurs.

En effet, au cours des trois essais précédents, notre estimateur présente une erreur d'estimation de la concentration inférieure à 1%. La prochaine étape de validation consiste donc à vérifier le comportement de l'estimateur sur des données produites par le système réel.

### 3.3.9 Application sur des données capteur

Afin de valider le comportement de l'estimateur sur des données réelles, nous avons utilisé des données fournies par *MEAS-France / TE Connectivity* correspondant à des essais effectués en cryostat sur des capteurs *UQS* en version B.

#### Considérations techniques

Notre estimateur étant prévu pour fonctionner à une fréquence d'échantillonnage de 1Hz, et les données fournies étant de fréquences variables, comprises entre 0.2 et 0.5Hz, nous avons du effectuer une interpolation entre les points pour produire des données utilisables par notre algorithme.

Les bruits de mesure configurés dans l'estimateur étant valables pour un échantillonnage direct, l'estimateur s'est cependant retrouvé dans l'incapacité de rejeter automatiquement les «glitch» (sauts de signaux, dus à un parasite ou à une erreur lors de l'enregistrement des données), ce qui nous a amené à devoir retirer manuellement les données erronées avant de les fournir à l'estimateur.

En effet, une donnée erronée se retrouvait répétée sur plusieurs échantillons au travers de l'interpolation mise en place, ce qui désorientait le comportement de l'algorithme. Cette intervention n'impacte cependant pas la représentativité du test car une fois embarqué, l'estimateur sera directement alimenté, à fréquence fixe, par les mesures du capteur.

#### Protocole de test

Pour ce test, le produit subit deux jeux de mesures successifs :

- une phase de calibration, au cours de laquelle le produit est immergé dans de l'urée à 33%, et subit quatre paliers de températures :  $0^{\circ}C$ ,  $20^{\circ}C$ ,  $40^{\circ}C$  et  $60^{\circ}C$  (Figure 3.36),
- une phase de test, au cours de laquelle le produit est immergé dans de l'urée à 32.5% et subit les quatre mêmes paliers de température.

Sur le premier jeu de données, l'estimateur est positionné en mode calibration afin d'estimer les gains de transmission individuels des LEDs. Sur le second jeu de données, l'estimateur est positionné en mode estimation d'urée afin de tester la fonctionnalité première du produit.

#### Étape de calibration

Comme décrit précédemment, l'étape de calibration a pour but de compenser les dispersions au niveau des gains du produit à partir de mesures effectuées dans un état connu. La Figure 3.37 présente les valeurs des gains estimés par l'estimateur de calibration lors de cette procédure sur les données du capteur.

Nous remarquons ainsi que, contrairement à la même procédure effectuée lors de la simulation et dont les résultats sont présentés en Figure 3.31, les gains varient au cours de la phase de calibration.

Ces variations peuvent traduire des variations environnementales, des modifications comportementales du produit ou une imprécision de modélisation. Si nous les recoupons avec le profil thermique de la phase de calibration (Figure 3.36), nous remarquons qu'ils correspondent aux paliers thermiques, ce qui privilégie l'hypothèse de l'imprécision dans le modèle.

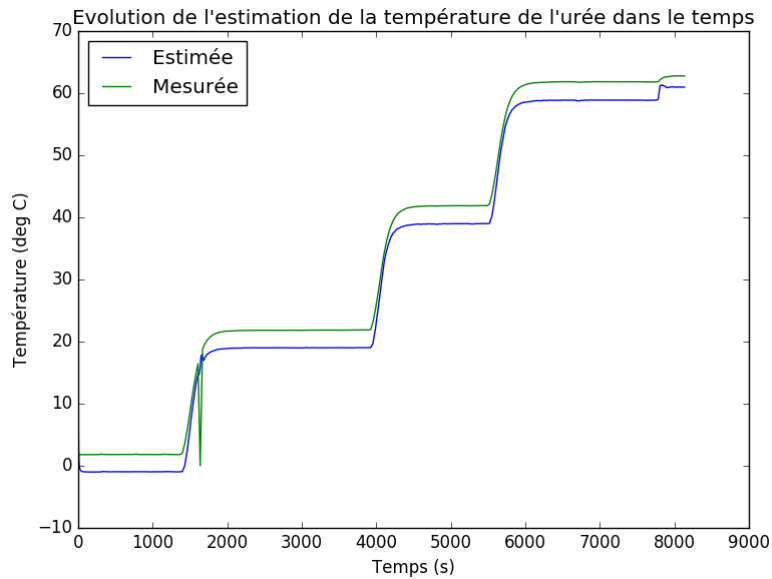


FIGURE 3.36 – Profil de température lors de la phase de calibration du capteur

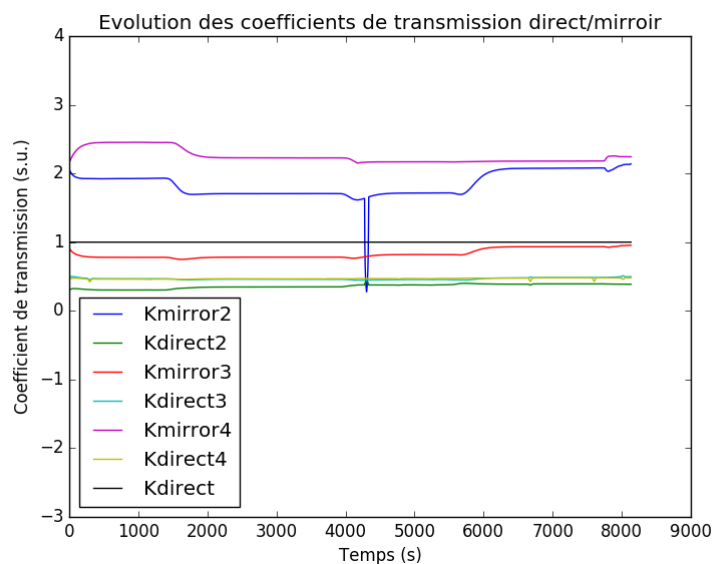


FIGURE 3.37 – Évolution des gains estimés lors de la calibration du capteur

Quoi qu'il en soit, une variation des gains au cours de la calibration pose la question de la valeur à fixer comme consigne pour l'estimateur de concentration. Dans le cas de notre simulation, nous avons fait le choix de mémoriser la dernière valeur de gain estimé (ce qui permet lors de l'intégration dans le produit de diminuer l'utilisation mémoire en n'ayant pas de sauvegarde de l'historique).

### Étape d'estimation de la concentration

Pour la phase d'estimation de la concentration, l'estimateur de concentration est calibré à partir des gains déterminés précédemment, puis se voit appliquer les jeux de mesures de la phase de test. La concentration du fluide est alors estimée (Figure 3.38).

Nous remarquons ainsi que la concentration est estimée entre 15% et 55%, avec une

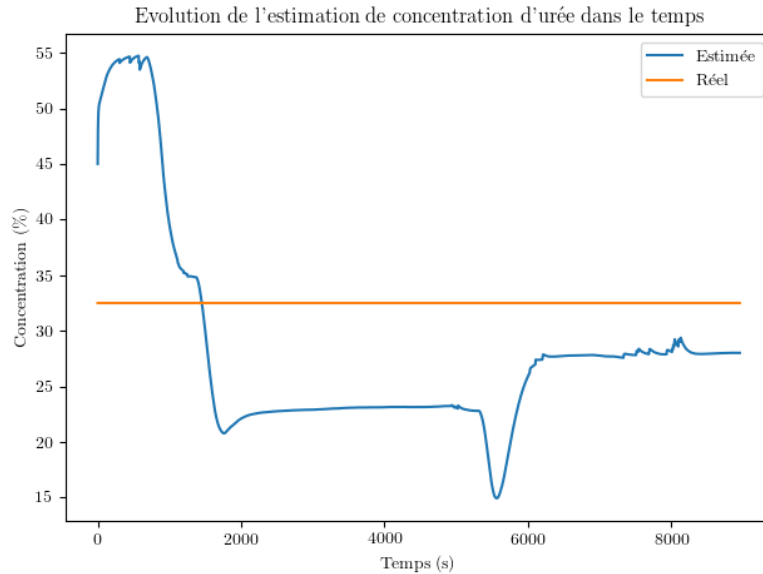


FIGURE 3.38 – Estimation de la concentration d'urée

concentration moyenne de 28.09% contre une concentration réelle de 32.5% et un écart-type de 9.32%.

Ces résultats confirment l'hypothèse effectuée lors de la calibration quant à l'imprécision du modèle du capteur.

### Interprétation

A partir des différentes simulations effectuées, nous remarquons que l'estimateur a un comportement prometteur lorsqu'il est appliqué sur le modèle de simulation, mais diverge lorsqu'il est appliqué sur les données produites par le système réel.

Ce comportement traduit donc une imprécision dans le modèle que nous avons utilisé pour représenter le système. Par exemple, une ou plusieurs des hypothèses prises lors de la modélisations ne sont pas valables, à savoir (non exhaustif) :

- La dissipation thermique est constante après allumage du capteur, et ne dépend pas de la température,
- Le comportement en absorption est linéaire en fonction de la concentration en urée,
- Les gains de transmission sont constants quelles que soient les conditions de fonctionnement du système,
- Les composants électroniques non optiques (amplificateurs, résistances, convertisseur analogique/numérique...) ne sont impactés que de façon négligeable par la température,
- Les dilatations, et autres comportements thermo-mécaniques, n'impactent pas le comportement du capteur...

Chacune de ces imprécisions peut impacter le comportement du modèle, et par la même occasion celui de l'estimateur, le rendant ainsi inopérant pour l'application sélectionnée.

### Pistes d'améliorations

D'après les résultats présentés ci-dessus, nous déduisons que l'estimateur n'est pas utilisable en l'état actuel sur le système réel. Nous pouvons cependant en tirer des pistes

de travail permettant de lever cette limitation.

La première piste consisterait à continuer d'affiner le modèle des éléments du capteur. Comme nous l'avons dit précédemment, certaines caractérisations (telle que celle des photo-diodes) ont été effectuées avec une faible précision et mériteraient d'être refaites pour correspondre plus précisément à la réalité du capteur.

De plus, les caractérisations prises en entrée du modèle ont été effectuées sur un seul composant et ne sont donc pas représentatives d'un point de vue statistique. Nous ne sommes en effet pas capable de déterminer la dispersion de certaines valeurs entre deux composants, ou au cours de la vie du composant.

Une seconde piste consisterait à ne plus utiliser un modèle complet du capteur mais plutôt à se baser sur un modèle boîte noire complet, faisant correspondre à chaque jeu de paramètres d'entrée, un jeu de mesures au travers de tables de recherches. Ce modèle pourrait alors être renseigné en phase de calibration de façon simple en faisant passer le produit dans les différents états et en sauvegardant les informations dans la mémoire.

L'estimateur de concentration utiliserait alors le modèle boîte noire de la même façon qu'il utilise actuellement le modèle orienté signal afin de déterminer la concentration d'urée. Nous détaillerons dans la section suivante les travaux effectués en ce sens.

### 3.3.10 Question de puissance

Bien qu'étant la solution la plus élégante, le modèle orienté signal poserait un problème de taille lors de son implémentation sur cible (dans l'hypothèse où il serait fonctionnel). Comme évoqué au cours du Chapitre 2, les problématiques de complexité algorithmique sont une partie centrale de la réalisation de systèmes embarqués. Bien que la complexité de l'estimateur de Kalman soit modérée si nous choisissons avec soin nos vecteurs d'états et de mesure, le modèle orienté signal présenterait quant à lui une complexité trop importante pour être intégré au sein de l'architecture actuelle.

Il faut en effet considérer que le système actuel est construit autour d'un micro-contrôleur 8bits, dépourvu de *FPU*, cadencé à 16MHz et ne disposant que de 128Ko de mémoire Flash (dont seulement 60Ko sont disponibles actuellement).

Or, d'un point de vue mémoire, les *LUT* utilisées pour représenter le modèle des différents éléments du système dépasseraient probablement l'espace disponible pour permettre une précision suffisante pour le système. De plus, chaque itération du modèle demande un grand nombre de multiplications ( $700 * 3 * 2 = 4200$ ), et ce répété pour chaque tirage de l'estimateur *HUKF*, ce qui est irréalisable à la fréquence d'échantillonnage de 1Hz.

A titre indicatif, sur un PC utilisant un processeur *Intel Core i7* à 3.5GHz, l'estimation du scénario de test de 8135s est effectuée en 400s, soit un temps moyen par échantillon de 50ms.

L'approche boîte noire est donc la piste privilégiée pour répondre à la problématique d'estimation sur ce capteur.

## 3.4 Approche sans modèle

Les approches sans modèle, ou «boîte noire» peuvent se présenter sous plusieurs formes :

- Absence complète de modèle formel, donc sous forme de *LUT*,
- Modèle hybride polynomial, cherchant à synthétiser le comportement complet du système sous la forme d'un polynôme.



La seconde solution étant la voie déjà explorée par *MEAS-France / TE Connectivity* sur le produit, nous nous concentrerons sur la version LUT, et plus particulièrement sur la mise à jour des données de la LUT au cours de la calibration.

### 3.4.1 Méthode des LUTs de calibration

Le point principal lors d'un système en boîte noire est la question de la calibration de ce dernier. En effet, dans le cadre d'un système comme *UQS*, le comportement du système dépend de la température des différents éléments de ce dernier. Cette variable ne pouvant pas être commandée sur le terrain, le capteur doit donc être en mesure «d'attendre» que la température soit bonne pour mémoriser l'information, et ce, à condition qu'elle soit fiable.

Une calibration polynomiale n'est alors possible que lorsque la quantité de données fiables acquises est suffisante, on peut alors effectuer une régression sur les données pour converger vers les valeurs des coefficients du polynôme.

Pour ces différentes raisons, nous avons choisi, en collaboration avec *Tom FOUGERE* de *MEAS-France / TE Connectivity*, de mettre en place un système de mémorisation des mesures sous forme de LUT. Lorsque celles-ci sont effectuées dans un fluide de référence, il devient alors théoriquement possible au produit de s'auto-calibrer s'il dispose de suffisamment d'informations.

#### Principe de fonctionnement

La température étant une information continue, et les points de mémorisation de la LUT étant des points discrets, nous nous sommes retrouvés face à la question de la stratégie à adopter pour mettre à jour les informations dans la LUT.

Notre choix s'est porté sur une méthode utilisant deux LUT différentes :

- une permettant de mémoriser l'information utile (par exemple, les niveaux de réception des LEDs en fonction de la température),
- l'autre permettant de donner un niveau de confiance dans les données contenues dans la première.

Ainsi, à partir de ces deux informations, nous pouvons effectuer une régression linéaire pondérée permettant d'estimer le polynôme représentant le mieux le comportement du système.

*Nous noterons que pour cette stratégie, nous n'utilisons plus qu'une seule information de température, et non les trois mesures présentées dans le modèle orienté signal.*

A chaque mesure, nous calculons ainsi un indice de confiance en la mesure, en se basant, entre autre, sur :

- la «distance thermique» : il s'agit de la distance entre température mesurée et chaque point de la LUT,
- la «distance de mesure» : il s'agit de la distance entre la mesure courante et la valeur correspondante dans la LUT,
- la «stabilité de mesure» : il s'agit de la différence entre la mesure courante et la mesure effectuée à la seconde précédente.

Ainsi, en se basant sur ces indices, nous pouvons déterminer un niveau de confiance en notre mesure. Ce niveau de confiance est alors utilisé pour mettre à jour la LUT de confiance et la LUT de mesures (en se basant sur le rapport entre la confiance de la mesure et la confiance précédente).

Donc, si la différence entre la calibration actuelle du produit et la mesure est importante, la confiance est faible et la calibration est faiblement modifiée. Cependant, si la mesure persiste de façon cohérente autour de la même valeur, la confiance en l'état de calibration diminue et la confiance en la mesure augmente, ce qui fait converger la calibration vers la mesure en cours. Plus les valeurs se rapprochent, plus la confiance augmente, pour finalement considérer le point comme très fiable.

De cette façon, si le système est soumis à des températures suffisamment variées durant un temps suffisamment long, la confiance en chaque point parcouru et mémorisé sera importante, et le produit pourra se calibrer pour compenser la dégradation de ses performances.

Par exemple, sur un profil thermique simple, et sur une seule LED, nous obtenons lors d'une simulation l'évolution de l'erreur normalisée (une erreur de 1 correspond à une erreur de 100%) présentée en Figure 3.39. Pour ce test, la LUT est composée d'une valeur tous les  $5^{\circ}C$  sur la plage  $-10^{\circ}C - 60^{\circ}C$  et la température parcourt la totalité de cette plage au cours du cycle.

Nous remarquons qu'au cours du temps, l'erreur (moyenne et écart-type) ainsi que l'erreur quadratique mesurées sur les valeurs, entre la table de référence simulée et la table de calibration estimée, tendent à diminuer au fur et à mesure que de nouvelles températures sont explorées.

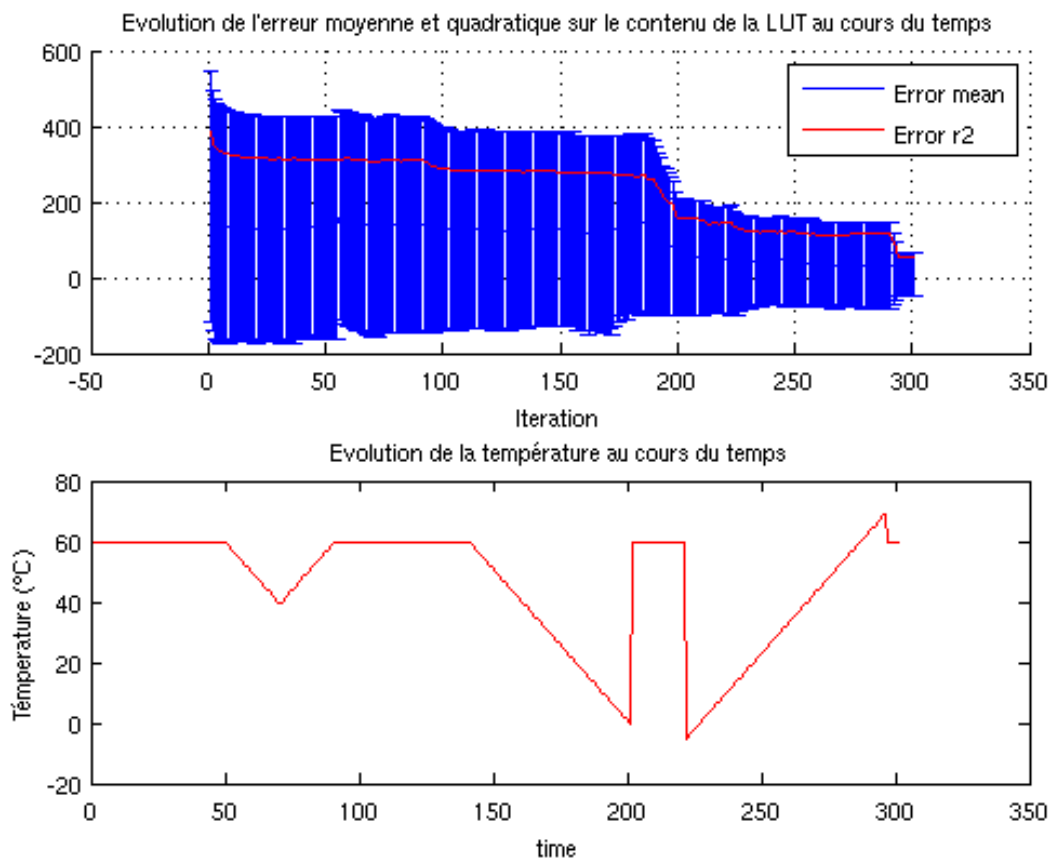


FIGURE 3.39 – Evolution de l'erreur au cours d'une simulation sur LUT

## Piste d'utilisation

A ce stade, les travaux menés sur cet algorithme n'ont pas été intégrés avec un estimateur permettant de déterminer la concentration d'urée. Il serait cependant possible, en dressant un modèle de simulation du système se basant sur les LUT ainsi que sur les gains directs/miroirs utilisés dans le modèle orienté signal du capteur, de réaliser un filtre *HUKF* similaire à celui présenté plus haut.

La seule condition d'utilisation est de garantir que la procédure de calibration est effectuée sur un seul plein d'urée, dont la concentration est parfaitement connue.

## 3.5 Conclusion

Au cours de ce chapitre, nous avons illustré avec un cas concret industriel notre méthodologie permettant d'estimer un paramètre physique en fusionnant les différentes mesures du capteur.

Dans un premier temps, nous nous sommes attardés sur la méthodologie orientée modèle, plus intuitive à mettre en place, qui nous a permis de comprendre en détails le comportement du capteur. Nous nous sommes cependant rendus compte que la caractérisation séparée des différents éléments était un processus difficile à mettre en oeuvre, et ne rendait finalement pas compte assez précisément du comportement réel du capteur assemblé pour être utilisable au sein d'un estimateur (Figure 3.40).

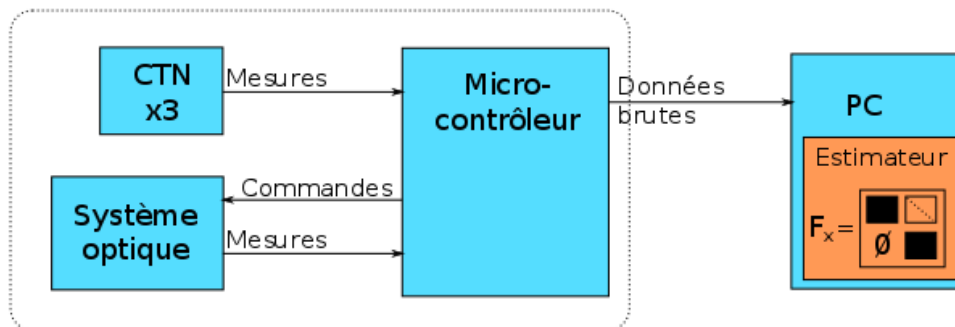


FIGURE 3.40 – Le micro-contrôleur n'est qu'une interface entre le PC et les éléments sensibles

Nous avons de plus remarqué que la charge de calculs nécessaire à embarquer pour mettre en place un estimateur autour d'un tel modèle était parfaitement irréaliste compte tenu des contraintes techniques liées à la cible du produit.

D'un point de vue conceptuel, la première étape correspond à dire que le capteur effectue les mesures physique et transmet les données brutes au calculateur de bord (ECU) (Figure 3.41).

Notre objectif étant de distribuer l'intelligence entre les différents éléments du véhicule, nous nous orientons plutôt vers une intégration de plusieurs petits filtres de Kalman simplifiés au plus près de la mesure, pour ne fournir à l'ECU que l'information utile (Figure 3.42).

Cependant, nous pourrions pousser encore plus loin le raisonnement, en implémentant les différents filtres de Kalman sur plusieurs unités de calculs (*Processing Unit*, PU) pouvant être des micro-contrôleurs ou des circuits logiques dédiés (Figure 3.43) : *Complex Programmable Logic Device* (CPLD), *Field Programmable Gate-Array* (FPGA), ou

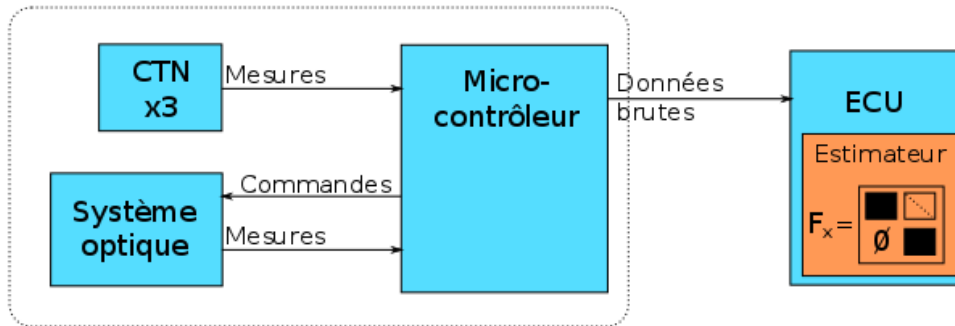


FIGURE 3.41 – Le calculateur de bord effectue le calcul et contient toute l’intelligence

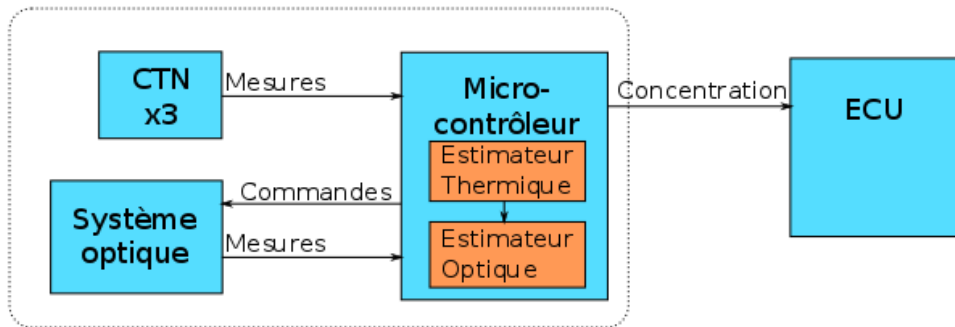


FIGURE 3.42 – Le micro-contrôleur effectue le traitement et ne transmet que l’information utile à l’ECU

*Application-Specific Integrated Circuits (ASIC).*

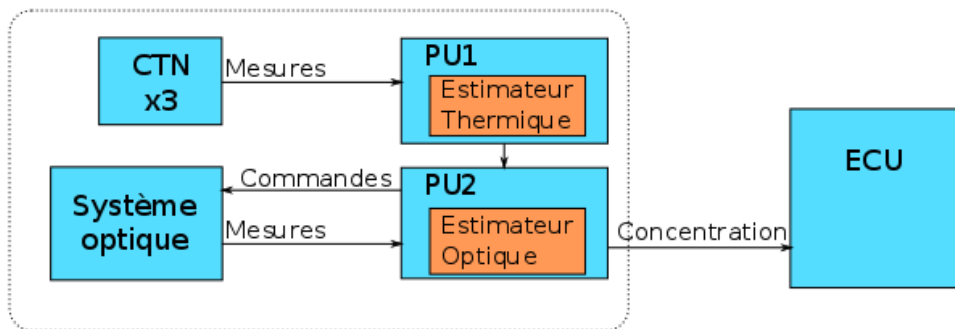


FIGURE 3.43 – La complexité est répartie sur différentes unités de traitements intégrées au capteur

Cette méthodologie a été explorée au cours de nos travaux sur le projet *Architecture Reconfigurable Embarquée Multi-capteurs (AREM)* [14] que nous explorerons au cours du chapitre 4.

Cette étude nous a cependant permis de valider la méthodologie au niveau des estimateurs, qui présentent de bons résultats sur les simulations et une charge de calculs acceptable, à condition de changer de modèle pour passer sur un modèle de LUT, nettement plus rapide à itérer.

Nous avons de plus proposé une stratégie permettant de construire ce modèle de façon dynamique et embarquée, de sorte que le capteur soit en mesure de s’auto-calibrer lorsqu’il est dans des conditions de fonctionnement connues.

Bien que ces travaux n'aient pas eu le temps d'aboutir sur un démonstrateur complet, rassemblant ces différents blocs, nous sommes confiants quant à la faisabilité et aux résultats pouvant être fournis par un tel système, en embarqué et temps réel.

Afin d'adresser plus en profondeur cette problématique de temps-réel, nous explorons au cours du chapitre suivant l'applicabilité de cette méthode sur les problématiques de temps-réel plus strict sur des cibles embarquées.

# Bibliographie

- [1] Arnaud Frobert Jean Milpied, Olivier Lepreux. Sensitivity of scr control strategies to diesel exhaust fluid quality : a simulation study. 2015.
- [2] Martyn V. Twigg. Urea-scr technology for denox after treatment of diesel exhausts. *Fundamental and Applied Catalysis* :221–232, 2014.
- [3] M. Stavros A. SKARLIS. *IR spectroscopy based kinetic modeling of NH<sub>3</sub>-SCR on Fe-zeolites : Application for diesel engines aftertreatment system simulation*. PhD thesis, 2013.
- [4] M.W. Markus. Optimisation of the snscr-denox method using diode laser spectroscopy. 5 :115–115, 2003.
- [5] Donald BURNS ; Emil CIURCZAK. *Handbook of Near-Infrared Analysis, Third Edition (Practical Spectroscopy)*. pp. 349–369. 2007.
- [6] Hoeil Chung ; Mark A. Arnold ; Martin Rhiel ; David W. Murhammer. Simultaneous measurements of glucose, glutamine, ammonia, lactate, and glutamate in aqueous solutions by near-infrared spectroscopy. *Applied Spectroscopy*, 50 :270–276, 1996.
- [7] Shih-Shin Tseng Tai-Sheng Yeh. A low cost led based spectrometer. *Journal of the chinese chemical society*, 53 :1067–1072, 2006.
- [8] J.B. Cumming. Temperature dependence of light absorption by water. *Nuclear Instruments and Methods in Physics Research Section A : Accelerators, Spectrometers, Detectors and Associated Equipment*, 713 :1–4, 2013.
- [9] A. A. Efremov ; N. I. Bochkareva ; R. I. Gorbunov ; D. A. Lavrinovich ; Y. T. Rebane ; D. V. Tarkhin ; Y. G. Shreter. Effect of the joule heating on the quantum efficiency and choice of thermal conditions for high-power blue ingan/gan leds. *Semiconductors*, 40 :605–610, 2006.
- [10] Otto Pursiainen ; Norbert Linder ; Arndt Jaeger ; Raimund Oberschmid ; Klaus Streubel. Identification of aging mechanisms in the optical and electrical characteristics of light-emitting diodes. *Applied Physics Letters*, 79, 2001.
- [11] David L. Allara. Aging of polymers. *Environmental Health Perspectives*, 11 :29–33, June 1975.
- [12] Richard STEVENS. *UNIX Network Programming, Volume 2, Second Edition : Interprocess Communications*. Prentice Hall.
- [13] C.T. Kelley. *Iterative Methods for Optimization*. Frontiers in Applied Mathematics, 1999.
- [14] Aurélien Valade et al. Système intelligent multi-capteurs dédié à la capture du mouvement. *CETISIS*, pages 155–160, 2014.



# Chapitre 4

## Applicabilité de la méthode avec contrainte temps-réel

Au cours du chapitre précédent, nous avons montré qu'il était possible, à partir d'un modèle suffisamment détaillé, de concevoir et de réaliser un capteur embarqué en utilisant la fusion de données multi-physiques à partir d'un filtre de Kalman.

Les travaux présentés, reposant uniquement sur des données transmises au travers d'un bus CAN, et non directement traitées sur le composant posent encore la question de l'applicabilité en temps-réel d'une telle méthodologie. Pour adresser cette problématique, nous allons nous pencher sur un second projet réalisé au cours de cette thèse : le projet *Architecture Reconfigurable Embarquée Multi-capteurs* (AREM) [1, 2].

L'objectif de ce projet est de concevoir et de réaliser un réseau type BodyLAN à partir de systèmes embarqués à base de centrales inertielles, pour permettre le monitoring temps réel du mouvement sportif (du point de vue biomécanique).

Puis d'apporter une analyse instantanée des données mesurées, pour évaluer la qualité du mouvement et le comparer à un "mouvement expert".

Enfin, cet ensemble multi capteur doit apporter un retour temps réel sur la qualité du mouvement effectué soit pour un athlète ou encore pour un entraîneur. Pour cela, l'une des informations pouvant être utile serait l'orientation d'un capteur dans l'espace, ce qui permet de déterminer l'orientation d'un segment du corps de l'athlète et donc pourrait permettre la reconstruction du squelette, ou de plusieurs segments stratégiques en multipliant le nombre de capteurs sur le corps (principe de la *motion capture* [3]). En équipant chacune de ces centrales de son propre système de traitement, en faisant ainsi une architecture distribuée, nous avons décidé d'effectuer la fusion de données permettant d'obtenir l'orientation d'un capteur dans l'espace, à partir des données brutes, de façon locale sur chaque unité de traitement. Nous ne nous concentrerons ici que sur le fonctionnement d'une seule de ces unités.

Nous allons dans un premier temps généraliser la méthode appliquée pour le capteur UQS, puis l'appliquer au capteur AREM, en nous attardant sur la problématique du temps-réel. Ensuite, nous présenterons la solution que nous avons choisie pour mettre en place ce système, puis nous présenterons les résultats obtenus à partir de l'un de ces capteurs. Finalement, nous proposerons un modèle plus contraint permettant de rajouter de l'information, et de tendre vers un indicateur de performances en temps réel.



## 4.1 Généralisation de la méthode

En se basant sur ces travaux, nous pouvons généraliser la méthode pour l'appliquer sur différents types de capteurs destinés à être embarqués avec des ressources limitées, en suivant la procédure suivante :

- Dresser la liste des entrées/sorties et des paramètres d'influence du système,
- Définir, si nécessaire, les paramètres de calibration du système,
- Déterminer les différents cas d'utilisation de ce système, ainsi que les conditions liées,
- A partir de là, déterminer les vecteurs de commande, de mesure et d'état du système dans les différents cas d'utilisation,
- Si nécessaire, séparer les paramètres indépendants sous la forme de plusieurs sous-systèmes,
- Mettre en place un modèle (mathématique ou basé sur des LUTs) permettant de déterminer les sorties du système en fonction de son état interne et de ses commandes,
- Mettre en place un ou plusieurs filtres de Kalman (linéaire, non-linéaire ou hybride) permettant d'estimer l'état interne du système à partir du modèle.

Afin de mieux ajuster les paramètres des filtres de Kalman, il est intéressant de disposer de jeux de mesures sur un nombre de capteurs suffisamment large pour en tirer des valeurs statistiques de référence (valeurs moyenne, dispersion, bruits, vieillissement...).

## 4.2 Contraintes matérielles et temporelles

La capture et l'analyse du mouvement humain, et particulièrement lors du geste sportif, nécessite la capacité de traiter les données fournies par les capteurs à une cadence importante (pour des mouvements compris entre 1Hz et 10Hz selon les études [4, 5]). Dans notre cas, les capteurs fournissant des mesures à 50Hz (soit un temps de traitement maximal par échantillon de 20ms), nous avons sélectionné cette valeur pour contrainte.

Le matériel choisi est de plus un capteur *System on Package*, le *iNemo-M1* [3] de chez *ST Microelectronics*, composé de :

- Un accéléromètre 3 axes et magnétomètre 3 axes *LSM303DLHC*,
- Un gyroscope 3 axes *L3GD20*,
- Un micro-contrôleur 32bits *STM32F103REY6*, sans *FPU*, cadencé à 72MHz.

Pour des raisons d'ergonomie, d'encombrement et de capacité des batteries, nous avons fait le choix d'effectuer la totalité du traitement, ainsi que la gestion de la communication avec l'environnement extérieur au travers d'un lien WiFi, au sein même du micro-contrôleur intégré au capteur.

Nous obtenons ainsi le système présenté en Figure 4.1.

## 4.3 Application de la méthode au système

En appliquant la méthodologie proposée sur ce système pour déterminer l'orientation d'un capteur à partir des mesures, nous obtenons les étapes suivantes (Figure 4.2).

### Liste des entrées/sortie et paramètres d'influence

Ainsi, nous avons par exemple :

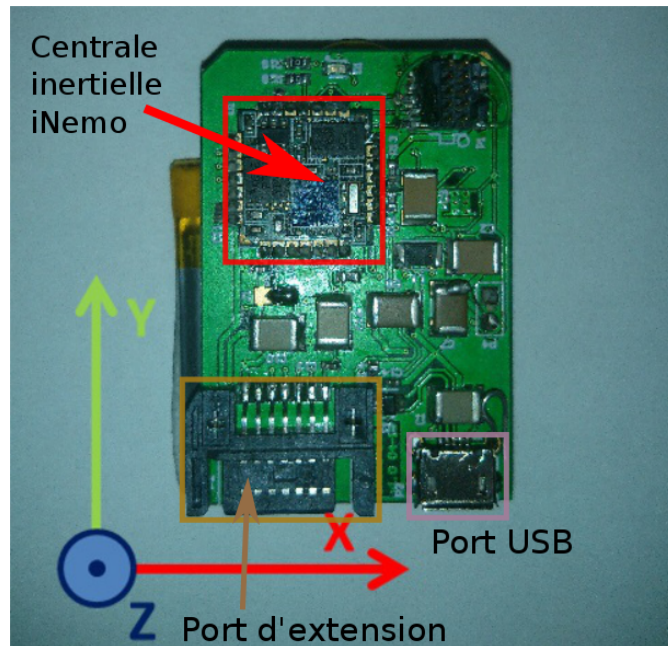


FIGURE 4.1 – Système AREM intégré

- Pour entrée du système, son *orientation dans l'espace*, que nous pouvons représenter en utilisant les angles d'Euler pour définir la rotation du capteur par rapport au repère NED (*North East Down*),
- Pour paramètres d'influence, le *champ magnétique environnant* (perturbations électromagnétiques, champ magnétique terrestre), ainsi que *son accélération* (accélération gravitationnelle et déplacement du capteur),
- Pour sorties du système, les mesures, selon les axes  $\vec{x}$ ,  $\vec{y}$  et  $\vec{z}$  du capteur, de l'accélération, de la vitesse de rotation et du champ magnétique environnant.

### Paramètres de calibration

Sur des capteurs de centrales inertielles intégrés à faible coût (tels que ceux utilisés au sein du projet AREM, ou dans les téléphones portables actuels), nous avons à compenser lors de la calibration :

- les *offsets* de mesure sur les accéléromètres et gyroscopes,
- les gains de mesure sur les accéléromètres et gyroscopes,
- le champ magnétique résiduel (*Hard/Soft Iron Distortion*).

Ces compensations doivent être effectuées lors d'une phase de calibration du capteur.

### Cas d'utilisation

Concernant les cas d'utilisation, le capteur sera utilisé dans les cas suivants :

- une phase de calibration, au cours de laquelle le capteur est déplacé selon toutes les rotations autour des axes  $\vec{x}$ ,  $\vec{y}$  et  $\vec{z}$  pour déterminer les gains et offsets de mesures, et ceci à partir des valeurs min/max et moyennes des mesures,
- une phase d'estimation d'attitude, au cours de laquelle nous chercherons à déterminer l'orientation du capteur à partir de ses mesures et de sa calibration.

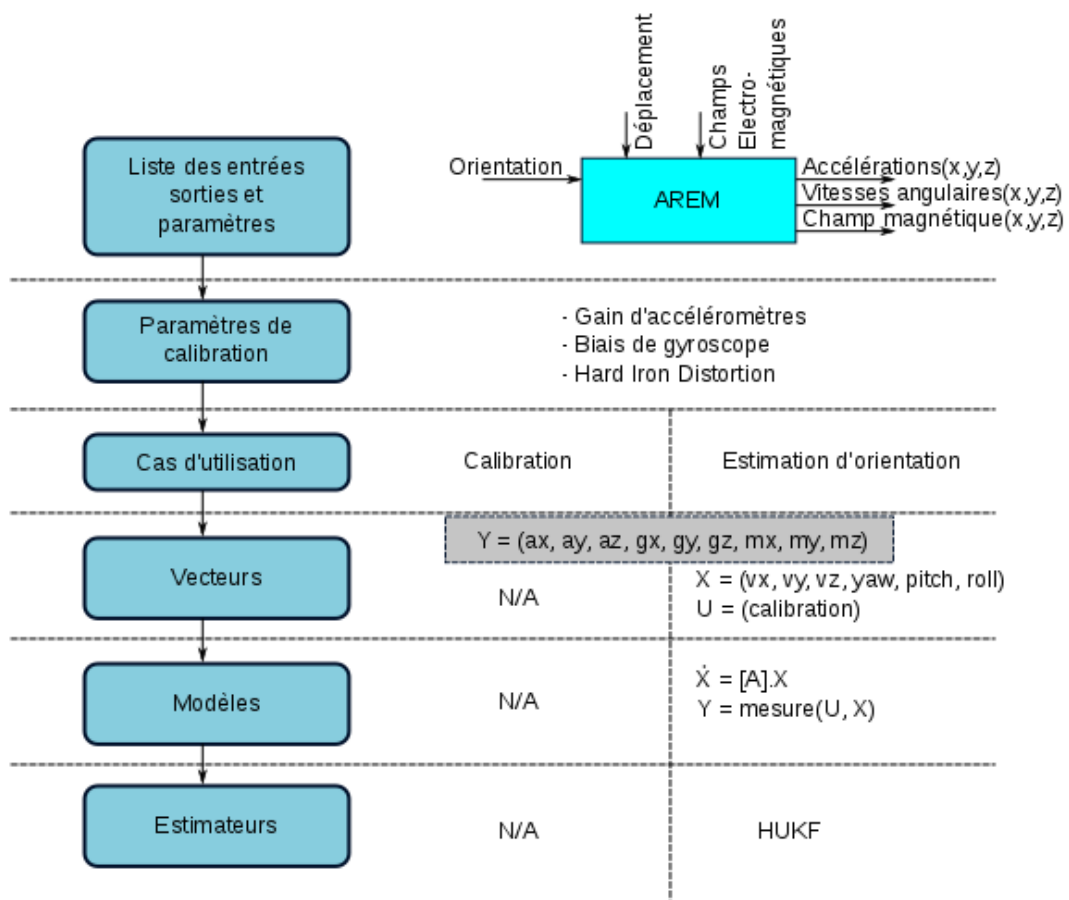


FIGURE 4.2 – Méthodologie appliquée à AREM

## Vecteurs du modèle

Dans tous les cas, le vecteur de mesure du capteur est constitué des sorties de celui-ci, à savoir ici les valeurs d'accélération, de vitesse de rotation et de champ magnétique sur les axes  $\vec{x}$ ,  $\vec{y}$  et  $\vec{z}$ , que nous noterons  $a_x, a_y, a_z, g_x, g_y, g_z$  et  $m_x, m_y, m_z$ . Ainsi le vecteur de mesure du capteur sera alors :

$$\mathbf{Y} = \begin{pmatrix} a_x \\ a_y \\ a_z \\ g_x \\ g_y \\ g_z \\ m_x \\ m_y \\ m_z \end{pmatrix} \quad (4.1)$$

Au cours de la phase de calibration, seules les valeurs minimales, maximales et moyennes de ces mesures nous intéressent, il n'est donc pas utile de mettre en place de filtre de Kalman, et donc de vecteurs associés.

Au cours de la phase d'estimation, le vecteur de commande  $\mathbf{U}$  est composé des gains et offsets de calibration. Nous pouvons avoir par exemple les biais  $b_{mx}, b_{my}$  et  $b_{mz}$  pour le magnétomètre,  $b_{gx}, b_{gy}$  et  $b_{gz}$  pour le gyroscope et les gains  $g_{ax}, g_{ay}$  et  $g_{az}$  pour l'accéléromètre, ce qui donne le vecteur :

$$\mathbf{U} = \begin{pmatrix} g_{ax} \\ g_{ay} \\ g_{az} \\ b_{gx} \\ b_{gy} \\ b_{gz} \\ b_{mx} \\ b_{my} \\ b_{mz} \end{pmatrix} \quad (4.2)$$

Le vecteur d'état pourrait quant à lui être composé des positions et des vitesses angulaires selon les angles d'Euler :

$$\mathbf{X} = \begin{pmatrix} v_x \\ v_y \\ v_z \\ yaw \\ pitch \\ roll \end{pmatrix} \quad (4.3)$$

## Mise en place du modèle

A partir de ces informations, nous pouvons dresser le modèle mathématique de projection des vecteurs de gravité et de nord magnétique qui permettront de projeter les mesures. On aura donc une fonction d'évolution pouvant être linéaire :

$$\frac{\delta yaw}{\delta t} = v_x \quad (4.4)$$

$$\frac{\delta pitch}{\delta t} = v_y \quad (4.5)$$

$$\frac{\delta roll}{\delta t} = v_z \quad (4.6)$$

$$(4.7)$$

$$\dot{\mathbf{X}} = [\mathbf{A}] \mathbf{X} + [\mathbf{B}] \mathbf{U} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \mathbf{X} + \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \mathbf{U} \quad (4.8)$$

et une fonction de mesure non linéaire :

$$\mathbf{Y} = \text{measure}(\mathbf{U}, \mathbf{X}) \quad (4.9)$$

## Mise en place du filtre de Kalman

A partir du modèle précédent, il suffit de mettre en place un filtre de Kalman hybride pour la phase d'estimation de l'orientation du capteur pour déterminer l'orientation de celui-ci (tel que présenté en 1.2.5).

Nous avons donc un vecteur d'état de 6 états, un vecteur de mesure de 9 mesures et un vecteur de commande de 9 commandes. A partir de là, nous pouvons calculer la complexité du filtre de Kalman associé qui serait de 9309 FLOPs par estimation, sans optimisation particulière du code d'estimation. A une fréquence de 50Hz, cela fait un besoin calculatoire de 465 450 FLOPs/sec. Or, nous estimons d'après le tableau 2.1 (Chapitre 2) que le STM32F103 à 72MHz a une capacité de calcul de 340 000 FLOPs/s !

Avec optimisation du code, et en particulier des algorithmes d'inversion de matrice, nous pouvons cependant espérer diminuer à 7 851 FLOPs par estimation. Nous avons aussi la possibilité de n'effectuer l'étape de correction que pour un échantillon sur deux, ce qui permet de diminuer suffisamment la complexité pour rendre l'opération envisageable. Cette solution particulière est celle adoptée par *ST Microelectronics* pour sa bibliothèque *iNemo Engine*[6].

L'un des inconvénients de la représentation *yaw*, *pitch*, *roll* utilisée ici est la présence de singularité par blocage de cardant [7]. L'une des solutions à ce problème est l'utilisation de la représentation des rotations par un quaternion [8][9].

De tels filtres ont déjà été étudiés, optimisés et documentés [3][10]. Nous nous sommes donc appuyés sur ces travaux pour ce projet avec l'utilisation de l'application *iNemo*

*Engine Lite* [11][12] de chez *ST Microelectronics*, qui effectue la fusion *Attitude Heading Reference System (AHRS)* [13] de données des différents capteurs de la centrale inertielle pour déterminer l'attitude du système en utilisant les quaternions comme représentation interne.

Sur cette application, le filtre est en mesure d'effectuer la fusion de données, sur le micro-contrôleur cadencé à 72MHz, à une fréquence de 50Hz (soit 20ms entre chaque estimation), et fournit un résultat d'estimation corrigée à 25Hz (soit 40ms entre chaque résultat fournit) (Figure 4.3), tout en gérant la sauvegarde de données sur carte SD et la retransmission au travers d'un lien WiFi en temps réel, permettant ainsi un affichage ou un post-traitement des orientations du capteur.

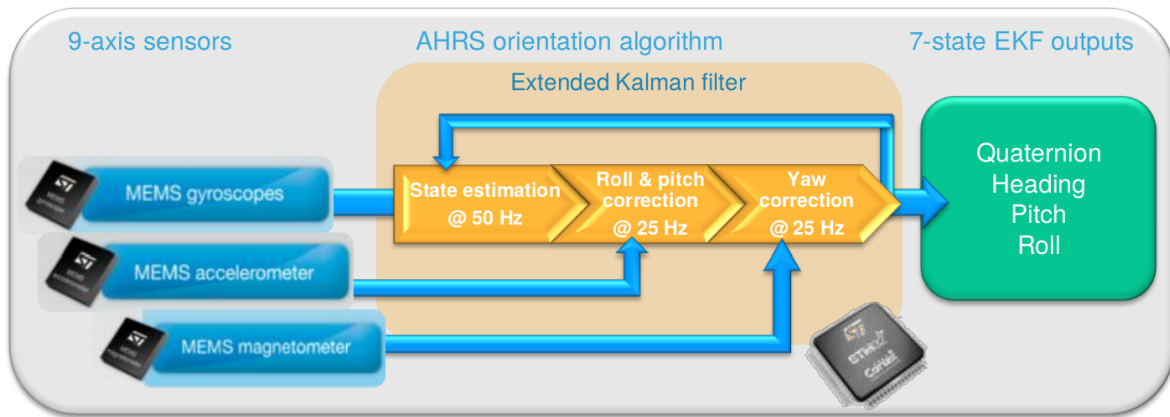


FIGURE 4.3 – Architecture EKF de la bibliothèque *iNemo Engine* [6]

### 4.3.1 Essais en environnement contrôlé

Afin de valider le fonctionnement du système, nous avons testé celui-ci en environnement contrôlé, afin de comparer sa mesure avec celle d'un système Vicon servant de référence. Pour cela, nous avons installé le capteur sur un support solidaire de trois capteurs passifs Vicon [14] (Figure 4.4). Cet assemblage a ensuite été positionné sur un plan de référence et tourné successivement selon les axes X, Y et Z.

Le système de référence est composé de douze caméras *Vicon MX3+* et du système de traitement associé. Il fournit la position de chacun des capteurs passifs dans l'espace, à une fréquence de 200Hz. A partir des coordonnées de ces 3 points, nous reconstruisons l'orientation dans l'espace du support de référence du capteur, que nous pouvons ensuite comparer à la mesure effectuée par le système AREM.

Afin de comparer les mesures des deux systèmes, nous avons tracé les orientations mesurées par les deux systèmes, en utilisant les angles d'Euler comme représentation, en interpolant les données acquises par AREM afin d'obtenir un nombre d'échantillons compatible entre les deux systèmes. Les résultats obtenus sont présentés en Figure 4.5.

Nous remarquons ainsi que l'allure générale du mouvement est correctement retranscrite, cependant la dynamique n'est pas parfaitement identique. Ce phénomène est en partie dû à des pertes de données lors de la transmission de l'information entre la carte AREM et le PC durant la phase de mesure. En effet, la principale différence entre les deux graphes se trouve au niveau de l'axe des X qui passe par un point de singularité sur le Vicon (lorsque l'angle selon l'axe des Z atteint les  $90^\circ$ ). Concernant l'exploitation du

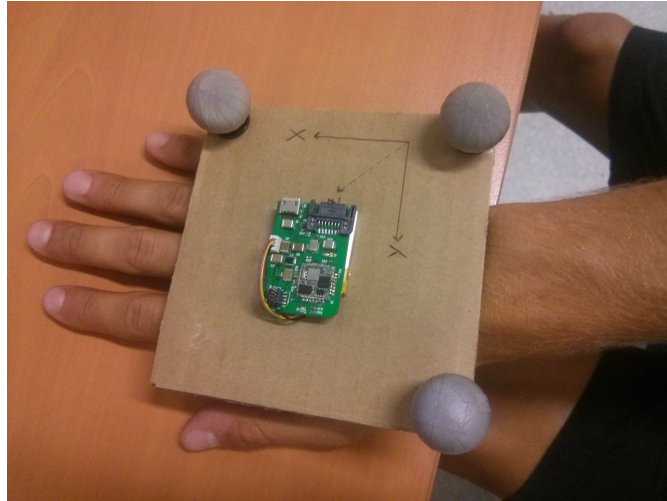


FIGURE 4.4 – Installation du capteur AREM conjointement aux capteur passifs Vicon

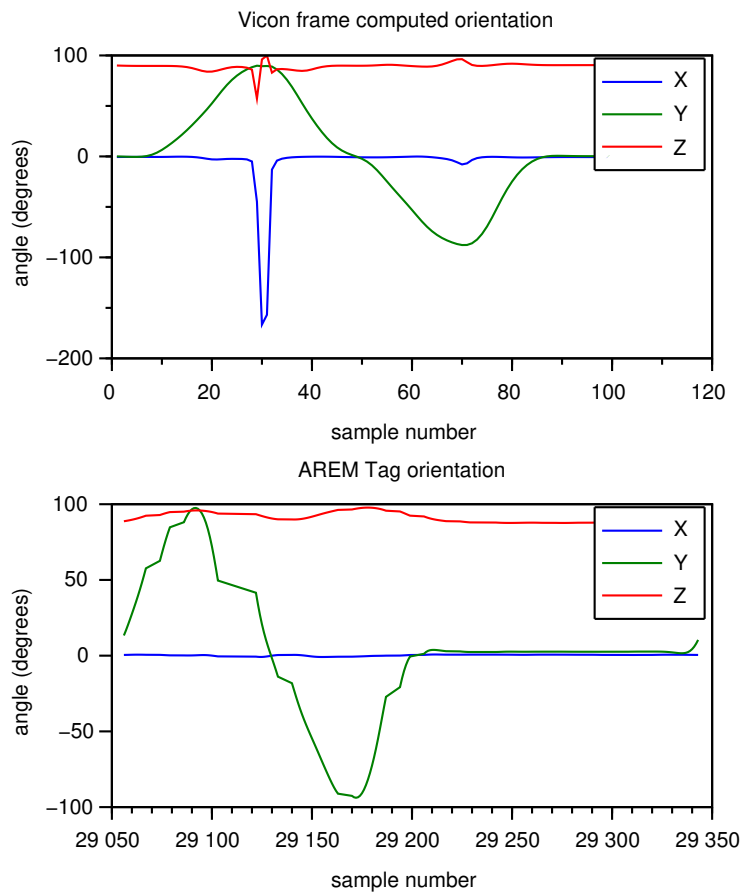


FIGURE 4.5 – Comparaison de l'orientation mesurée par le système Vicon et par AREM sur une rotation selon l'axe Y

mouvement en Y, nous observons bien des fréquences similaires du fait de la différence d'origine de l'axe du temps et de la différence de fréquence d'échantillonnage.

Afin de palier à ces problèmes de communication, les versions suivantes du système ont été équipées d'une carte SD permettant de garder un stockage local des résultats d'estimations, ce que nous avons utilisé pour l'étude menée sur le mouvement en natation.

### 4.3.2 Utilisation pour l'analyse de la nage

L'objectif premier du système AREM étant la qualification du mouvement sportif, nous avons effectué des essais sur le mouvement de natation. Pour cela, nous avons installé une carte AREM sur la main gauche d'un nageur (Figure 4.6) et collecté l'évolution de l'orientation de ce capteur au cours d'une session de natation [15].

Des études similaires ont été menées par [16] et [17].

Au cours de ces essais, nous avons constaté que le lien WiFi était rompu à partir du moment où la main du nageur était immergée, l'eau agissant comme un filtre absorbant la bande de fréquence de 2.4GHz du WiFi. Les données présentées lors de cette étude sont donc collectées depuis la carte SD intégrée au prototype du capteur.

Partant de là, nous avons effectué une campagne de mesures sur différents nageurs, de niveau amateur à élite, afin de comparer le mouvement mesuré et de rechercher des indicateurs de performance (lié à la distinction de niveau et/ou à la dégradation du mouvement par la fatigue [15]). Nous obtenons par exemple les Figures 4.7a et 4.7b sur un nageur amateur et un nageur élite.

Depuis ces données, nous pouvons par exemple remarquer que le nageur amateur a une fréquence de nage plus importante que le nageur élite, la forme globale du geste est de plus relativement différente. Autant d'indicateurs qui peuvent être étudiés de façon approfondie pour, à terme, permettre la mise en place d'un diagnostic embarqué en temps-réel lors du geste sportif.

La poursuite de ces travaux permettrait, en collaboration avec des spécialistes en biomécanique, de fournir aux coachs sportifs, pour les sportifs de haut niveau ou amateurs, des indications quantifiables leur servant au quotidien dans leur pratique, et le tout instantanément lors de l'action.

L'intérêt ici est de montrer qu'en appliquant ces méthodes, nous sommes désormais en mesure d'effectuer de la mesure d'indicateurs de mouvement en temps-réel et en embarqué, ce qui ouvre la voie, dans les années à venir à des applications plein air telles que l'aide à l'entraînement sportif, là où, il y a peu de temps, les environnements contrôlés tels que le Vicon étaient rois.

## 4.4 Conclusion

Au cours de ce chapitre, nous avons montré que la méthode présentée sur le capteur UQS, avec des contraintes temps-réel faibles, pouvait être généralisée et appliquée avec des contraintes de temps plus exigeantes sur une application complètement différente telle que l'analyse du mouvement sportif, en temps réel et en conditions non contrôlées.

Cette étude va dans le sens des objets connectés intelligents qui sont amenés à être de plus en plus présents dans notre environnement dans les années à venir. Nous pouvons citer par exemple les capteurs connectés aux smartphones tels que les cardio-fréquencemètres, ainsi que les montres connectées permettant de disposer de l'information instantanément, en temps réel, d'un seul coup d'oeil.

L'utilisation de filtrage avancé prédictif tel que le filtre de Kalman nous permettrait ainsi d'enrichir par un modèle, et donc d'améliorer la réponse de capteurs, afin d'éviter d'obtenir des erreurs. En l'occurrence, un capteur d'activité de *Smart Watch* nous indiquant que l'on a marché 10km durant nos heures de bureau alors que nous sommes simplement restés installés devant notre ordinateur.



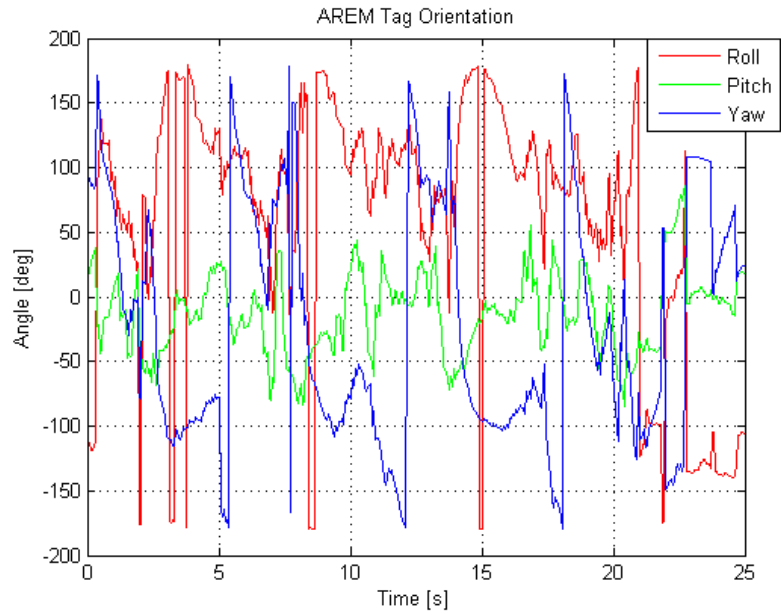


(a) Vue de dessus

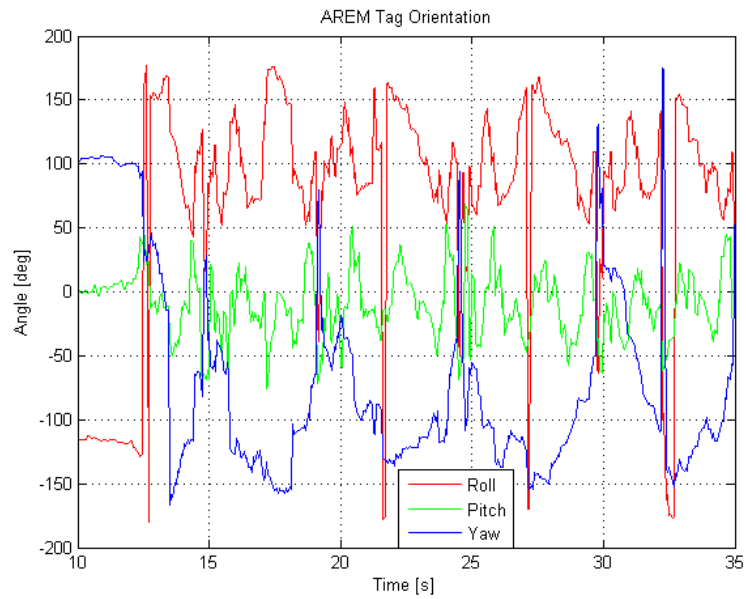


(b) Vue de côté

FIGURE 4.6 – Installation du système AREM pour le test en piscine



(a) Nageur amateur



(b) Nageur élite

FIGURE 4.7 – Comparaison du mouvement d'un nageur amateur avec un nageur élite



# Bibliographie

- [1] Aurélien Valade et al. Système intelligent multi-capteurs dédié à la capture du mouvement. *CETISIS*, pages 155–160, 2014.
- [2] A. VALADE et al. Embedded sensors system applied to wearable motion analysis in sports. *9th International Conference on Biomedical Electronics and Devices*, 2016.
- [3] Carmen M. N. Brigante ; Nunzio Abbate ; Adriano Basile ; Alessandro Carmelo Faulisi and Salvatore Sessa. Towards miniaturization of a mems-based wearable motion capture system. *IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS*, 58(8), August 2011.
- [4] Kenneth A. Mann ; Frederick Werner ; Andrew K. Palmer. Frequency spectrum analysis of wrist motion for activities of daily living. *Journal of Orthopaedic Research*, 7 :304–306, 1989.
- [5] Qinggang Meng ; Baihua Li ; Horst Holstein. Recognition of human periodic movements from unstructured information using a motion-based frequency domain approach. *Image Vision Computing*, 24 :795–809, 2006.
- [6] ST Microelectronics. inemo engine lite, overview and download instructions.
- [7] David Hoag. Apollo guidance and navigation considerations of apollo imu gimbal lock. *MIT Instrumentation Laboratory Document*, E-1344, 1963.
- [8] Jack B. Kuipers. *Quaternions and rotation Sequences : a Primer with Applications to Orbits, Aerospace, and Virtual Reality*. Princeton University Press, 1999.
- [9] Simon L. Altmann. *Rotations, quaternions, and double groups*. Oxford University Press, 1986.
- [10] Sebastian O.H. Madgwick. An efficient orientation filter for inertial and inertial/magnetic sensor arrays, April 2010.
- [11] ST Microelectronics. Um1601 user manual, 2015.
- [12] ST Microelectronics. Um1679 user manual, 2013.
- [13] Mei Wang ; Yunchun Yang ; R.R. Hatch ; Yanhua Zhang. Adaptive filter for a miniature mems based attitude and heading reference system. *Position Location and Navigation Symposium*, 2004.
- [14] Werner Goebel and Caroline Palmer. Temporal control and hand movement efficiency in skilled music performance. *PLoS One*, 2013.
- [15] M. Mangin et al. An instrumented glove for swimming performance monitoring. *IC Sports 2015*, pages 53–58, 2015.
- [16] Andrew J Callaway, Jon E Cobb, and Ian Jones. A comparison of video and accelerometer based approaches applied to performance monitoring in swimming. *International Journal of Sports Science & Coaching*, 4(1) :139–153, 2009.

- [17] Ethos : Miniature orientation sensor for wearable human motion analysis. *SENSORS*, 2010.

# Conclusion

Notre objectif au début de ces travaux consistait à étudier des méthodes pouvant être adaptées pour améliorer la fiabilité des mesures sur un capteur embarqué, en se basant sur un ensemble de mesures multi-physiques. Avec pour application principale le produit UQS de notre partenaire *MEAS-France / TE Connectivity*, et pour application secondaire le capteur AREM développé au *LAAS-CNRS* au cours de cette thèse.

Après quelques recherches, nous avons arrêté notre choix sur l'utilisation d'estimateurs de Kalman permettant de déterminer de façon optimale l'état du système, pour peu que nous disposions d'un modèle adéquat de celui-ci. Nous étudions en détail le fonctionnement de tels filtres pour les systèmes linéaire et non-linéaires au cours du chapitre 1.2. Ainsi, nous montrons qu'en fonction du type du système, nous choisissons plutôt :

- un filtre de Kalman pour les systèmes linéaires. En effet, ce dernier est le plus simple à itérer et est parfaitement adapté pour de tels systèmes,
- un filtre de Kalman étendu (EKF) pour les systèmes présentant des non-linéarités faibles (avec des comportements d'évolution et de mesure représentés par des fonctions monotones),
- un filtre *Unscented Kalman Filter* (UKF) pour les systèmes présentant de fortes non-linéarités, pouvant entraîner une divergence de l'estimation sur un filtre EKF, ou lorsque les ressources de calculs ne sont pas limitées,

Nous présentons de plus une optimisation hybridant les filtres entres eux pour les systèmes disposant, par exemple, d'une évolution linéaire et d'une mesure non-linéaire.

Une fois cette méthode établie, nous étudions en détails la complexité calculatoire de l'utilisation de tels filtres au cours du chapitre 2, en la mettant en parallèle avec les puissances de calculs généralement constatées dans les marché des systèmes embarqués. Cette information nous permet d'en déduire directement la complexité maximale d'un système pouvant être implémentée dans une puce en fonction des performances de celle-ci et du temps de calcul disponible pour chaque estimation.

A partir de là, notre travail a consisté en l'établissement d'une méthodologie applicable sur le capteur UQS, pouvant être généralisée sur d'autres capteurs, et d'en déterminer les limites.

## 4.5 Histoire d'un aller et retour : application et réajustement de la méthode

Nous avons illustré notre méthode au cours du chapitre 3 sur l'application industrielle du projet UQS et montré qu'elle permet d'intégrer un ensemble de mesures qui, prises individuellement nécessiteraient un environnement de mesure contrôlé (comme en laboratoire) pour en extraire l'information utile, mais qui, lorsqu'elles sont considérées dans leur ensemble, permettent de compenser les variations environnementales pour en extraire une

mesure enrichie.

Pour cela, nous nous appuyons sur un modèle du système, qui doit être réalisé afin de déterminer les relations entrées/sorties en fonction des différents paramètres d'influence du système. Ensuite, nous listons les grandeurs pouvant être directement obtenues depuis les mesures (températures...) ainsi que celles qui doivent être estimées (concentration d'urée...).

Au besoin, nous avons la possibilité d'ajouter des variables d'ajustement permettant de compenser des dispersions entre produits (étape de calibration, illustrée par les gains direct sur UQS), ou des éléments pouvant varier dans le temps, et pouvant avoir un impact sur le comportement des mesures (cas des gains miroir, reflétant une variation de la puissance d'émission ou du gain direct commun permettant de compenser l'impact de poussières dans le chemin optique).

Une fois le modèle dressé, nous pouvons mettre en place un estimateur permettant de déterminer les paramètres inconnus (non mesurables directement). Afin que le résultat d'estimation soit le moins biaisé possible, il est important de limiter les colinéarités dans le modèle qui ont pour conséquence de détériorer l'observabilité de celui-ci (*cf.* chapitre 1.2.3). Cette dernière peut être facilement déterminée dans le cadre d'un système linéaire par observation des matrices du modèle, elle est cependant nettement plus complexe à obtenir, précisément en ce qui concerne un système non linéaire car elle varie en fonction de l'état du système et de la jacobienne de celui-ci (pour un système non linéaire, certains états peuvent être moins observables que d'autres, conduisant donc à une mesure moins précise dans cette zone).

Avec nos travaux sur le projet UQS, nous avons remarqué que, bien que plus intuitif à mettre en place, un modèle du système uniquement basé sur l'agrégation des comportements physiques individuels de chacun de ses éléments pouvait se trouver complètement inadapté pour une implémentation dans un système embarqué dont les ressources de calculs sont limitées. Dans le cadre du projet UQS, ceci est dû au fait que la propagation des spectres optiques et des modifications associées implique un grand nombre de calculs pour chaque itération du modèle, ainsi qu'une forte occupation mémoire, ce qui, couplé à une puissance de calcul faible, rend l'estimation impossible en temps réel.

A partir de ce constat, nous avons préféré concentrer notre effort sur une solution hybride, dont le modèle de fonctionnement du capteur serait représenté par une LUT de mesures en fonctions des paramètres d'entrées. Cette LUT, pouvant être mise à jour de façon dynamique lorsque les conditions sont contrôlées, peut ensuite être couplée à un estimateur de Kalman non linéaire ou hybride (type EKF, UKF, HEKF ou HUKF) pour déterminer l'état du système une fois calibré. Cette solution présente à la fois un avantage de simplicité et de flexibilité, ce qui la rend privilégiée dans le cadre d'un système embarqué, par définition limité en ressources.

Afin de garantir une certaine cohérence des données contenues dans la LUT, ainsi qu'une extrapolation sur les points n'ayant pas été mesurés, il est également possible d'effectuer une régression polynomiale pondérée sur les données stockées dans la LUT. Le résultat de cette régression pouvant servir directement de modèle comportemental du système, utilisé par l'estimateur.

## 4.6 Du diagnostique en temps réel

Après cela, nous présentons au cours du chapitre 4 l'application de la méthode sur le système AREM imposant des contraintes de temps-réel plus importantes. Pour ce système,

l'objectif est de déterminer en temps réel un indice de performances sportive à partir de mesures effectuées sur des données provenant d'une centrale inertielle 9 axes (composée d'un accéléromètre 3 axes, un gyroscope 3 axes et un magnétomètre 3 axes). Pour cela, nous choisissons de reconstruire l'orientation du capteur dans l'espace à partir de la fusion des données brutes produites par les centrales intérielles.

Pour les résultats soient pertinents d'un point de vue qualification du mouvement, nous désirons effectuer l'estimation à une fréquence de 50Hz, et nous disposons pour cela d'un microcontrôleur 32bits STM32F103 cadencé à 72MHz.

Nous proposons un modèle permettant de déterminer l'orientation du capteur dans l'espace, sous hypothèse que celui-ci n'effectue que des rotations et ne subisse donc pas d'accélération extérieures. Après estimation de la complexité algorithmique de notre modèle, nous constatons que celui-ci ne peut être implémenté tel-quel dans le système en satisfaisant la contrainte de 50Hz. Nous proposons alors différentes optimisations algorithmiques permettant de résoudre ce problème.

Finalement, nous présentons la solution que nous avons implémentée, reposant sur la bibliothèque proposée par *ST Microelectronics*, ainsi que les résultats obtenus lors de mesures effectuées, d'abord en conditions contrôlées, nous permettant de comparer le comportement avec un système *Vicon* utilisé comme référence, puis dans un environnement applicatif sur des nageurs.

## 4.7 Ouverture

Avec la croissance continue des puissances de calcul embarquées et la réduction des coûts des éléments sensibles, la fusion de données multi-physiques et les capteurs intelligents sont amenés à se multiplier dans nos environnements au cours des prochaines années. Ces mesures enrichies et pré-traitées seront alors plus fiables et aisées à utiliser dans des applications plus complexes telles que la robotique, la domotique ou le coaching.

La méthodologie que nous avons explorée au cours de ce mémoire nous fournit ainsi un point de départ permettant d'appréhender les enjeux liés au design de tels capteurs, se basant sur des approches ne pouvant être appliquées, jusqu'à il y a peu, qu'en post-traitement ou sur ordinateurs, pour obtenir des performances suffisantes pour garantir un temps-réel.

Il est cependant à noter que la complexité des systèmes allant aussi croissante, tous les systèmes ne pourront pas être résolus uniquement au travers de cette méthode. Si nous prenons par exemple le cas d'un système présentant de forts couplages colinéaires entre les différents paramètres, ou ne disposant pas de suffisamment de points de mesures pour le rendre observable, notre méthodologie devient inopérante.

Dans de tels cas, elle permet cependant de déterminer, au travers du modèle, quelles sont les inconnues à lever au travers de mesures supplémentaires, ainsi que d'établir une première estimation de la complexité calculatoire nécessaire à la résolution du modèle, et ainsi d'orienter la conception matérielle du système, qu'il s'agisse de l'ajout d'éléments sensibles, ou de dimensionnement de la puissance de calcul.

Concernant le modèle, nous préconisons d'identifier clairement quels sont les comportements linéaires et non-linéaires, et si possible découpler dans des sous-systèmes isolés (s'il n'existe pas d'interactions bidirectionnelles entre les deux). Dans ce cas, la complexité algorithmique est nettement optimisée et l'identification de paramètres est simplifiée. Il est aussi intéressant de noter qu'un grand nombre de systèmes physiques présentent des



évolutions linéaires et des sorties non-linéaires, dans de tels cas, l'utilisation d'un filtre hybride est la plus optimale en terme de compromis précision/performances.

Au-delà d'un enjeu scientifique, nous sommes persuadés que la multiplication des capteurs et la fusion de données embarquées au plus proche de la mesure constitue l'avenir des systèmes embarqués.

# Ma production scientifique

## Conférences internationales avec comité de relecture

- Aurélien Valade, Antony Costes, Anthony Bouillod, Morgane Mangin, Georges Soto-Romero, Jean-Yves Fourniols, Frederic Grappe. Embedded sensors system applied to wearable motion analysis in sports. *BioDevices* : 170-175, 2016
- M. Mangin, A. Valade, A. Costes, A. Bouillod, P. Acco and G. Soto-Romero. An Instrumented Glove for Swimming Performance Monitoring. *ICSports* : 53-58, 2015
- Juan D. Lemos, A. Mauricio Hernández, Georges Soto-Romero, Aurelien Valade. Instrumented Glove for Skills Assessment in Neurosurgical Simulation. *BioCAS*, 2014
- A. Valade, G. Soto Romero, C. Escriba, A. Bouillod, J. Pinot, J. Cassirame, Jean-Yves Fourniols, F. Grappe. Wearable multi-sensor system for embedded body position and motion analysis during cycling. *World Congress of Cycling Science*, 2014
- A Bouillod, J Pinot, A Valade, J Cassirame, G Soto-Romero, F Grappe. Gross efficiency is improved in standing position with an increase of the power output. *World Congress of Cycling Science, J Cycling Vol. 3(2)* : 6, 2014

## Conférences nationales avec comité de relecture

- Aurélien Valade, Anthony Bouillod, Florian Girard, Steeven Flores, Fabrice Severac, Georges Soto-Romero, Christophe Escriba, Jean-Yves Fourniols, Julien Pinot, Frederic Grappe, Juan Diego Lemos, Mauricio Hernandez. Système Intelligent Multi-capteurs dédié à la capture du mouvement. *CETISIS* : 155-160, 2014