



# Recursive identification of continuous-time systems with time-varying parameters

Arturo Padilla

## ► To cite this version:

Arturo Padilla. Recursive identification of continuous-time systems with time-varying parameters. Automatic. Université de Lorraine, 2017. English. NNT : 2017LORR0119 . tel-01647164

**HAL Id: tel-01647164**

**<https://theses.hal.science/tel-01647164>**

Submitted on 24 Nov 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.



## AVERTISSEMENT

Ce document est le fruit d'un long travail approuvé par le jury de soutenance et mis à disposition de l'ensemble de la communauté universitaire élargie.

Il est soumis à la propriété intellectuelle de l'auteur. Ceci implique une obligation de citation et de référencement lors de l'utilisation de ce document.

D'autre part, toute contrefaçon, plagiat, reproduction illicite encourt une poursuite pénale.

Contact : [ddoc-theses-contact@univ-lorraine.fr](mailto:ddoc-theses-contact@univ-lorraine.fr)

## LIENS

Code de la Propriété Intellectuelle. articles L 122. 4

Code de la Propriété Intellectuelle. articles L 335.2- L 335.10

[http://www.cfcopies.com/V2/leg/leg\\_droi.php](http://www.cfcopies.com/V2/leg/leg_droi.php)

<http://www.culture.gouv.fr/culture/infos-pratiques/droits/protection.htm>

# Identification récursive de systèmes continus à paramètres variables dans le temps

## THÈSE

présentée et soutenue publiquement le 05 Juillet 2017

pour l'obtention du

**Doctorat de l'Université de Lorraine**

(Spécialité Automatique)

par

Arturo PADILLA

### Composition du jury

|                      |                  |                                                                                        |
|----------------------|------------------|----------------------------------------------------------------------------------------|
| <i>Rapporteurs :</i> | Fouad GIRI       | Professeur, GREYC Lab<br>Université de Caen Basse-Normandie, Caen                      |
|                      | Alina VODA       | Maître de Conférences<br>GIPSA Lab, Grenoble                                           |
| <i>Examineurs :</i>  | Michel MENSLER   | Docteur, Technocentre Renault<br>Guyancourt                                            |
|                      | Thierry BASTOGNE | Professeur, CRAN<br>Université de Lorraine, Nancy                                      |
| <i>Invités :</i>     | Peter YOUNG      | Professeur Emeritus<br>Lancaster University<br>Systems and Control Group, Lancaster UK |
| <i>Directeur :</i>   | Hugues GARNIER   | Professeur, CRAN<br>Université de Lorraine, Nancy                                      |



*To my family.*



## Acknowledgements

The Ph.D. is a long way to go and in this way I have met a lot of people that I would like to thank. First I would like to sincerely thank my supervisor Hugues Garnier and Peter Young for their advices and constant support. Their help has been crucial to complete my Ph.D. My profound gratitude goes also to Juan Yuz, who put me in contact with the research group in Nancy and with whom I have had the pleasure to collaborate with before and during the Ph.D.

In the lab, I had the great opportunity to discuss with other professors and to collaborate with some of them. In that regard, I would like to thank Gérard Bloch, Florian Collin, Vincent Laurain, Gilles Millerioux, Marion Gilson, Éric Gnaedinger, Jean-Luc Noizette and Frederic Sarry.

During these three years in Nancy I met in the lab a lot of colleagues, unfortunately, some of them were there only for a few months. I am most grateful to them for the pleasant atmosphere in the lab and conversations we shared. I want to thank deeply to Mahmoud Abdelrahim, Marharyta Aleksandrova, Asma Barbata, Fengwei Chen, Laura Ciesa, Tejaswinee Darure, Thibaud Foltzer, Quentin Grandemange, Sabra Hamza, Paul Huynh, Jean Kuchly, Dawid Machala, Sarnavi Mahesh, Tatiana Manrique Espindola, Carlo Nainer, Fetra Rasoanarivo, Mathieu Rupp, Cyrine Selma, Krishnan Srinivasarengan and Sebastian Theron.

I want also to thank Brandon Dravie for his friendship and constant help with paperwork in french. Some of the things that I admire from you is your naturalness and good will. After living alone for two years, I decided to move in with Yusuf Bhujwalla and Valentin Pascu. Living together was a great experience and it gave me the opportunity to know you better. Impossible to forget is the rich but fantastic food prepared by Valentin. Regarding domestic affairs where french was required, like calling the agency for problems with the cooker, Yusuf was the man. Thanks a lot guys for everything (specially the good jokes), I have learnt a lot from you. I am also most grateful to Valentin's and Yusuf's girlfriends Nicoleta Melnic and Christelle Dulon for the nice moments that we shared.

Outside the lab, I had a great time too. It was sometimes enjoying good food in a restaurant, listening nice music in a party or something else. In this small adventures I met Marylene Espinola, Patricia Espinola, Andrea Espinoza, Cezar Ferreira, Jean Martin Kyssama, Madelin Leyva, Andrey Myagkiy, Guillermo Pizarro, Silvia Robertson, Elise Vouriot, Georgiy Zadvitskiy. Many thanks to you for those nice moments.

During my stay in Nancy, I joined a group of friends coming mainly from Latin America. We called ourselves the Miercolitos, since we used to hang out mainly on Wednesday (Miércoles in spanish) during the week. Life without going to the bar is not life. This is something that its very clear in the minds of the members of the Miercolitos, that was composed by Adriana del Pilar Arriola, Aliz Barbosa, Vincent Bontemps, Alejandro Borroto, Jordan Cavalier, Magali de la Cruz Barrón, Claudia de Melo Sanchez, Mariana Diaz, Isabelle Dollander, Claudia Domjahn, Carmen Gonzalez, Estibaly Hernandez, Anne-Claire Huerta, Jannes Kinscher, Mariana Limones, Karen Mancera, Damian Marcual, Roberto Rama, Natalia Rebolledo, Thibaud Sauvageon, Cristina Valencia and Robert Valverde. Thanks very much to all of you for the unforgettable days that we

spent together.

When something was not on track, my friends in Chile and some other places, have been always there to give me a hand. Many thanks to you for your help and good vibes.

Finally I would like to express my profound gratitude to my family for their love and constant support. In particular, my greatest thanks go to my parents, my brother Rodrigo and my grandfather Edison. My little brother Nicolás, who rests in peace, is always in my thoughts and in my heart. I dedicate this thesis with love to him and the rest of my family.

# Résumé étendu

Les personnes qui travaillent dans le domaine des Sciences et de l'ingénierie s'intéressent à la compréhension des phénomènes naturels, et à cette fin, ils construisent habituellement des modèles qui représentent le système observé. Dans le domaine de la dynamique des fluides par exemple, l'intérêt est de comprendre comment modifier la conception d'un navire afin d'améliorer sa maniabilité. Si le navire est déjà construit, le problème pour un ingénieur pourrait être de concevoir un contrôleur pour guider automatiquement le navire. Un modèle mathématique de la dynamique du navire, prenant la forme d'un ensemble d'équations, servirait alors à cette fin [Fossen, 2002]. Une approche pour obtenir un modèle réside dans l'application des lois fondamentales de la Physique comme les lois de Newton ou les lois de Kirchhoff. La deuxième option pour la modélisation est l'identification du système, qui consiste essentiellement à construire un modèle à partir des mesures collectées de l'entrée et de la sortie du système. Cette thèse porte sur cette dernière approche.

## Identification du système

La procédure d'identification des systèmes est illustrée à par la figure 1, où nous pouvons voir les trois entités principales, qui sont :

- Le jeu de données. Il s'agit d'une collection de mesures de l'entrée  $u$  et de la sortie  $y$  du système  $\mathcal{S}$ , et est désigné par  $Z_N = u(t_k), y(t_k)_{k=1}^N$ , où  $N$  est le nombre d'échantillons. Notez que la sortie  $y$  est contaminée par  $v$  qui peut représenter un bruit de perturbation et/ou de mesures provenant du capteur.
- La structure de modèles. C'est un ensemble désigné par  $\mathcal{M}$  avec des structures de modèles candidates qui pourraient représenter le système [Ljung, 1999, p. 111]. Une structure de modèle est une expression mathématique, qui relie  $u$ ,  $y$  et  $v$ , et est paramétrée par un vecteur  $\theta \in \mathbb{R}^{n_\theta}$ .
- Une méthode d'identification. Compte tenu des deux premières entités, une méthode d'identification doit être choisie afin d'estimer le vecteur de paramètres  $\theta$  et, par conséquent, le modèle.

Les modèles estimés à l'aide de techniques d'identification du système peuvent être classés en modèle de type *boîte grise* ou *boîte noire* [Ljung, 1999]. Dans le premier, les lois fondamentales de la Physique servent à construire la structure du modèle, qui comprend des paramètres inconnus. Ces paramètres sont ensuite estimés à partir d'un jeu de données et d'une méthode d'identification. Dans le cas d'un modèle de type boîte noire, la structure et ses paramètres sont directement déterminés à partir des données mesurées et la méthode d'identification. En outre, le modèle peut être à temps discret (DT) ou continu (CT). Dans ce travail, nous nous intéressons aux modèles linéaires CT qui peuvent être de type boîte grise ou boîte noire.

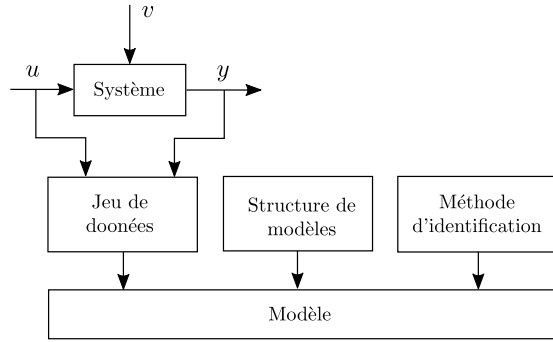


Figure 1: Entités de base pour l'identification d'un système: jeu de données, structure de modèles et méthode d'identification.

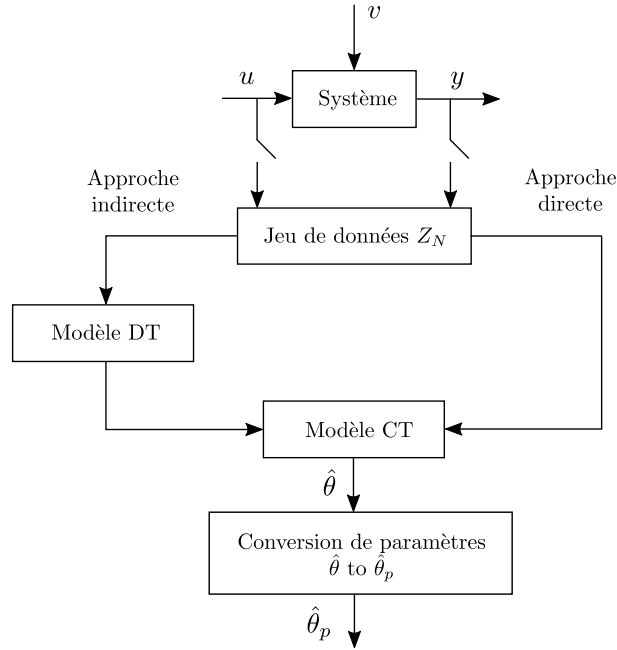


Figure 2: Approches indirectes et directes pour l'identification d'un modèle CT.

## Identification hors ligne des modèles à temps continu

Dans l'estimation hors ligne, également appelée identification non récursive, ou en bloc, l'ensemble des données doit être disponible a priori. Deux approches d'identification de modèles CT linéaires invariants dans le temps (LTI) peuvent être obtenus en utilisant une estimation non récursive résumée sur le schéma de la figure 2 [Garnier et al., 2008]:

- Approche indirecte. Tout d'abord, un modèle DT est identifié puis il est converti en CT. Dans le cas LTI, cela peut être fait dans MATLAB en utilisant la fonction `d2c`.
- Approche directe. Il vise à estimer le modèle CT directement à partir des données échantillonnées.

Comme le montre le schéma, les paramètres CT peuvent être convertis en paramètres physiques  $\hat{\theta}_p$ , si nécessaire. Il faut remarquer que quelle que soit l'approche retenue, directe ou indirecte, il faut faire une hypothèse sur la variation des signaux entre deux instants d'échantillonnage [Garnier et al., 2008].

Motivés par les avantages qu’offrent les méthodes directes (voir [Garnier et al., 2008, Garnier and Young, 2014]), nous avons décidé de travailler dans cette thèse avec ce type d’approche.

Différentes approches directes sont disponibles dans la littérature ; elles peuvent être classées comme suit [Garnier et al., 2003, 2008] :

- méthodes de filtrage linéaire
- méthodes intégrales
- méthodes des fonctions de modulation
- d’autres approches comme les méthodes de différence finie

Dans cette étude, nous considérons les méthodes de filtrage linéaire.

## Identification de modèles linéaires à paramètres variables dans le temps

L’identification des modèles CT linéaires à paramètres variables (LTV) partage les aspects méthodologiques avec le cas DT LTV. Par conséquent, après quelques commentaires sur les systèmes LTV, nous discutons des approches qui ont été proposées pour l’identification DT LTV. Nous terminons la section avec l’estimation des modèles CT LTV.

Un système linéaire à paramètres variables dans le temps peut être défini comme un système dont les propriétés varient au cours du temps. Dans la vie réelle, les processus sont généralement variables dans le temps, par exemple en raison des effets de vieillissement ou des changements dans les conditions environnementales telles que la température. Parfois, la durée d’observation du système qui nous intéresse est assez courte, ce qui permet de considérer le système comme invariant dans le temps. Cependant, ce n’est pas le cas pour les systèmes que nous traitons dans cette thèse. Les paramètres d’un système LTV peuvent évoluer selon les quatre manières suivantes [Niedźwiecki, 2000, p. 60] :

- variations constantes lentes (parfois appelés dérive de paramètres)
- variations brusques peu fréquentes
- variations selon les deux modes ci-dessus (changements lents et brusques)
- tous les autres types de variations (également appelés variations de paramètres rapides)

Dans ce travail, on suppose que les paramètres varient lentement dans le temps. En outre, nous restreignons notre attention sur les systèmes linéaires (LTV).

### Identification non récursive des modèles à temps discret

Les méthodes d’identification hors ligne peuvent également être utilisées pour l’identification des systèmes lentement LTV. Une approche simple et intuitive consiste à estimer des modèles LTI locaux à l’aide de la méthode des moindres carrés pondérés sur une fenêtre glissante. Deux options sont alors possibles [Niedźwiecki, 2000] :

- Les moindres carrés pondérés de façon uniforme (fenêtre glissante de type rectangulaire), impliquant que tous les poids sont égaux.

- Les moindres carrés pondérés de façon exponentielle, où l'influence des données anciennes est atténuée au moyen d'une pondération de type exponentiel.

L'identification des systèmes LTV peut également être effectuée à l'aide de modèles déterministes pour décrire les variations de paramètres. On peut ensuite réaliser l'estimation avec l'approche de la fenêtre glissante hors ligne. Une de ces techniques est exploitée par exemple dans [Chan and Zhang, 2011] via une modélisation polynomiale locale. Néanmoins, nous nous intéressons à l'identification en ligne, où il est difficile de trouver une représentation appropriée des paramètres variables à l'aide de modèles déterministes. En outre, le nombre de paramètres augmente, ce qui conduit à des valeurs estimées avec une variance plus grande.

## Identification récursive de modèles à temps discret

En estimation récursive, également appelée en ligne ou en temps réel, le modèle est mis à jour chaque fois que de nouvelles mesures sont disponibles. En comparaison avec l'estimation hors ligne, les algorithmes récursifs sont en général avantageux en termes de coût de calculs.

Un grand nombre de travaux de recherche a été réalisé sur l'estimation récursive des modèles DT (voir par exemple [Ljung and Söderström, 1983, Goodwin and Sin, 1984, Ljung and Gunnarsson, 1990, Ljung, 1999, Haykin, 2014] et les références s'y rapportant). Certains algorithmes bien connus, résumés dans [Ljung, 1999], sont :

- Méthode récursive des moindres carrés avec facteur d'oubli (RLS-FF). Cette approche simple est dérivée des moindres carrés pondérés de manière exponentielle. L'inconvénient de cette approche est qu'elle ne convient pas si les paramètres varient à des taux différents. Dans ce cas, un facteur d'oubli de type matriciel pourrait être utilisé comme cela a été suggéré dans [Saelid and Foss, 1983] (voir aussi [Niedźwiecki, 2000, p. 105]).
- Filtre Kalman (KF). Cet algorithme est obtenu en supposant que les paramètres varient comme une marche aléatoire, qui est un modèle stochastique. En tant que méthode basée sur le facteur d'oubli matriciel, l'avantage de KF est qu'il convient lorsque les paramètres varient à des vitesses différentes. La marche aléatoire est le modèle stochastique le plus simple de la classe de promenade aléatoire généralisée [Young, 2011], qui a une flexibilité accrue (voir aussi [Kitagawa and Gersch, 1984]).
- Méthode de la variable instrumentale récursive (RIV). En tant qu'alternative non récursive, cet algorithme est algébriquement très similaire au RLS, mais avec l'avantage qu'il possède des propriétés statistiques supérieures. Différentes variantes de RIV peuvent être considérées [Ljung and Söderström, 1983]. Pour suivre les variations de paramètres variables dans le temps, RIV peut être utilisé avec un facteur d'oubli ou le KF comme mécanisme d'adaptation [Young, 2011].
- Méthode d'erreur de prédiction récursive (RPED). Cet algorithme est une approche basée sur le gradient. La méthode RLS et les moindres carrés (LMS) font partie de cette famille. Notez que LMS et RLS sont tout aussi bons pour le suivi des paramètres à variabilité lente [Lindbom, 1995, p. 16].

L'estimation récursive peut également être réalisée à l'aide de modèles déterministes. Comme nous l'avons déjà mentionné, dans l'identification en ligne, il est difficile de trouver une représentation appropriée des paramètres variables dans le temps à l'aide de modèles déterministes. Néanmoins, la modélisation déterministe peut offrir de bonnes performances dans le cas de variations rapides des paramètres [Niedźwiecki, 2000]. Un

tel problème est abordé par exemple dans [Niedźwiecki and Klaput, 2002] en utilisant plusieurs modèles déterministes dans un schéma d'estimation parallèle. De cette façon, la difficulté de trouver un modèle déterministe est contournée. Les modèles déterministes sont également pris en compte dans le cas de certains modèles pour les systèmes non linéaires, tels que les paramètres dépendants de l'état (voir par exemple [Young et al., 2001, Young, 2011]) et les paramètres non linéaires variants (voir par exemple [Previdi and Lovera, 2004]). Enfin, il est également possible de combiner des modèles déterministes et stochastiques, comme dans le cas de la modélisation de la régression harmonique dynamique [Young et al., 1999, Young, 2011].

Les travaux de recherche pour l'estimation réursive des modèles LTV se sont poursuivis au cours des dernières décennies. Des approches basées sur la classe de modèles de promenade aléatoire généralisée sont présentées dans [Zheng and Lin, 2003] en considérant un algorithme basé sur RLS et dans [Rutström, 2005] en considérant un algorithme basé sur LMS. Un autre sujet de recherche récent est l'estimation de modèles non paramétriques dans le domaine temporel en utilisant une approche de régularisation / bayésienne (voir [Prando et al., 2016]).

Les méthodes d'identification réursive sont nombreuses et variées. Par conséquent, choisir une méthode appropriée n'est pas une tâche facile. Les critères pour un bon choix sont par exemple le taux de convergence, la capacité de suivi, la robustesse, l'implantation informatique et les propriétés numériques (voir plus dans [Haykin, 2014, p. 23]). Dans ce travail, nous proposerons des algorithmes basés sur RLS, KF et RIV.

## Identification réursive de modèles à temps continu

Une étude récente concernant l'identification des modèles DT et CT LTV est par exemple [Lataire, 2011], où une méthode fréquentielle hors ligne a été développée. Les techniques de sous-espace appliquées de manière réursive ont également été utilisées pour identifier les systèmes multivariables (MIMO) [Bergamasco and Lovera, 2011].

Les méthodes d'identification DT discutées ci-dessus pourraient être appliquées pour identifier un modèle de CT en utilisant l'approche indirecte. Une estimation réursive indirecte a été utilisée par exemple dans [Gustafsson and Olsson, 1999].

En ce qui concerne les approches directes, une option consiste à utiliser des méthodes intégrales (voir [Garnier et al., 2003]), comme celles utilisées dans [Jiang and Schaufelberger, 1991]. La méthode CT PEM réursive présentée dans [Östring and Gunnarsson, 2004] est également une méthode intégrale, et elle a été appliquée pour estimer les paramètres physiques d'un bras de robot flexible. Une autre classe de méthodes d'identification CT directe s'appuie sur le principe des fonctions de modulation. Une implémentation réursive utilisant des fonctions de modulation est proposée dans [Co and Ungarala, 1997, Ungarala and Co, 2000]. Une troisième classe d'approches d'identification CT directe est la méthode de filtrage linéaire. Par exemple, une approche basée sur un filtre linéaire en ligne est présentée dans [Dimogianopoulos and Lozano, 2001], où une estimation réursive utilisant les moindres carrés pondérés de manière exponentielle est exploitée. Les algorithmes réursifs, tels que RLS, RIV et le filtre de Kalman, peuvent être couplés avec des méthodes de filtrage linéaire [Åström and Wittenmark, 2008, p. 59]. Cela a été proposé par [Isermann and Münchhof, 2010], où la version réursive des moindres carrés associées à la méthode des filtres de variable d'état, est utilisée pour suivre non seulement les variations lentes mais également brusques de paramètres (voir aussi [Canudas de Wit, 1986]). Dans ce travail, nous nous concentrons sur des algorithmes réursifs basés sur des filtres linéaires, mais uniquement pour les systèmes CT dont les paramètres varient lentement dans le temps. De plus, nous développons des méthodes de variables instrumentales (IV) en raison de leurs propriétés statistiques supérieures.

## Organisation de la thèse et contributions

Dans le chapitre 2, l'objectif est d'étendre les méthodes IV à l'estimation des systèmes LTV fonctionnant à la fois en boucle ouverte et en boucle fermée. Grâce à des exemples numériques, nous montrons les bonnes performances générales de ces algorithmes.

En pratique, l'identification récursive des systèmes variant dans le temps est difficile, car plusieurs aspects pratiques et problèmes de mise en œuvre doivent être pris en considération. Dans les algorithmes récursifs pour les systèmes CT LTV que nous avons proposés au chapitre 2, des aspects pratiques et des problèmes d'implémentation similaires apparaissent. L'objectif du chapitre 3 est de présenter des solutions appropriées à certains d'entre eux. En ce qui concerne les aspects pratiques, nous abordons le problème de l'estimation des hyperparamètres des algorithmes présentés. Un autre aspect important est le problème de la qualité du signal d'excitation. Les problèmes de mise en œuvre qui mènent à l'amélioration de l'implantation numérique des algorithmes proposés sont également discutés.

Le chapitre 4 est dédié à deux applications. Le premier est un circuit électronique réel qui a récemment été proposé comme benchmark pour l'identification du modèle LPV et LTV [Lataire et al., 2015]. La seconde est une vanne papillon où l'on suppose que les paramètres du modèle varient en fonction de la température. Comme il existe des non-linéarités dans le système, la méthode d'identification proposée au chapitre 2 est alors adaptée à ce contexte.

# Contents

|                                                                    |            |
|--------------------------------------------------------------------|------------|
| <b>List of Abbreviations</b>                                       | <b>XI</b>  |
| <b>List of Symbols</b>                                             | <b>XII</b> |
| <b>1 Introduction</b>                                              | <b>1</b>   |
| 1.1 System identification . . . . .                                | 1          |
| 1.2 Off-line identification of continuous-time models . . . . .    | 2          |
| 1.2.1 Problem formulation . . . . .                                | 2          |
| 1.2.2 The indirect vs direct identification methods . . . . .      | 3          |
| 1.3 Identification of linear time-varying models . . . . .         | 7          |
| 1.3.1 Linear time-varying systems . . . . .                        | 7          |
| 1.3.2 Identification of discrete-time models . . . . .             | 8          |
| 1.3.3 Identification of continuous-time models . . . . .           | 12         |
| 1.3.4 Numerical example - Similar parameter variations . . . . .   | 16         |
| 1.4 Recursive identification in practice . . . . .                 | 19         |
| 1.5 Applications in recursive identification . . . . .             | 20         |
| 1.6 Organization of this thesis and contributions . . . . .        | 20         |
| 1.7 Publications . . . . .                                         | 21         |
| <b>2 Instrumental variable based methods</b>                       | <b>22</b>  |
| 2.1 Introduction . . . . .                                         | 22         |
| 2.2 Open-loop identification of LTI models . . . . .               | 23         |
| 2.2.1 Optimal IV estimation . . . . .                              | 23         |
| 2.2.2 Off-line IV algorithms . . . . .                             | 24         |
| 2.2.3 Recursive instrumental variable methods . . . . .            | 25         |
| 2.3 Open-loop identification of LTV models . . . . .               | 26         |
| 2.3.1 Problem formulation . . . . .                                | 26         |
| 2.3.2 Handling of the time-derivative issue . . . . .              | 27         |
| 2.3.3 Recursive instrumental variable methods . . . . .            | 28         |
| 2.3.4 Measures of estimator performance . . . . .                  | 30         |
| 2.3.5 Numerical example - Similar parameter variations . . . . .   | 31         |
| 2.3.6 Numerical example - Different parameter variations . . . . . | 33         |
| 2.4 Closed-loop identification . . . . .                           | 39         |
| 2.4.1 Problem formulation . . . . .                                | 39         |
| 2.4.2 Optimal off-line IV estimation of LTI models . . . . .       | 40         |
| 2.4.3 Off-line IV algorithms of LTI models . . . . .               | 40         |
| 2.4.4 Recursive estimation of LTI models . . . . .                 | 41         |
| 2.4.5 Recursive estimation of LTV models . . . . .                 | 42         |
| 2.4.6 Numerical example . . . . .                                  | 42         |
| 2.5 Conclusions . . . . .                                          | 44         |

|          |                                                                       |           |
|----------|-----------------------------------------------------------------------|-----------|
| <b>3</b> | <b>Practical aspects and implementation issues</b>                    | <b>46</b> |
| 3.1      | Aspects related to the SVF . . . . .                                  | 47        |
| 3.1.1    | Normalized vs ordinary SVF . . . . .                                  | 47        |
| 3.1.2    | Choice of $\lambda_{\text{svf}}$ . . . . .                            | 48        |
| 3.1.3    | Digital implementation of CT filtering operations . . . . .           | 48        |
| 3.2      | Aspects related to the IV approaches . . . . .                        | 49        |
| 3.2.1    | Start of the IV methods . . . . .                                     | 49        |
| 3.2.2    | Stability test . . . . .                                              | 50        |
| 3.2.3    | Correlation between filtered instruments and filtered noise . . . . . | 51        |
| 3.3      | Estimation of the normalized covariance matrix $Q_n$ . . . . .        | 51        |
| 3.3.1    | Review of maximum likelihood method . . . . .                         | 51        |
| 3.3.2    | Estimation of $Q_n$ using maximum likelihood . . . . .                | 53        |
| 3.3.3    | Numerical example . . . . .                                           | 54        |
| 3.4      | Handling of input/output initial conditions . . . . .                 | 56        |
| 3.4.1    | Problem formulation . . . . .                                         | 56        |
| 3.4.2    | Recursive least squares state-variable filter method . . . . .        | 56        |
| 3.4.3    | Numerical example . . . . .                                           | 58        |
| 3.5      | Numerical issues . . . . .                                            | 61        |
| 3.5.1    | Conditioning in LSSVF . . . . .                                       | 61        |
| 3.5.2    | Signal scaling in LSSVF . . . . .                                     | 63        |
| 3.5.3    | Signal scaling in RLSSVF . . . . .                                    | 64        |
| 3.5.4    | Review of Potter's algorithm . . . . .                                | 64        |
| 3.5.5    | Signal scaling in RSRIVC . . . . .                                    | 66        |
| 3.6      | Anti-windup techniques . . . . .                                      | 66        |
| 3.6.1    | Anti-windup techniques in discrete-time identification . . . . .      | 67        |
| 3.6.2    | Conditional updating . . . . .                                        | 68        |
| 3.6.3    | Levenberg-Marquardt regularization method . . . . .                   | 69        |
| 3.6.4    | Numerical example . . . . .                                           | 70        |
| 3.7      | Conclusions . . . . .                                                 | 72        |
| <b>4</b> | <b>Applications</b>                                                   | <b>75</b> |
| 4.1      | Benchmark data from an electronic bandpass filter . . . . .           | 75        |
| 4.1.1    | Problem formulation . . . . .                                         | 76        |
| 4.1.2    | Recursive estimation of the electrical circuit . . . . .              | 78        |
| 4.1.3    | Illustration of the signal scaling approach . . . . .                 | 79        |
| 4.2      | Electronic throttle control system . . . . .                          | 81        |
| 4.2.1    | Electronic throttle body model . . . . .                              | 82        |
| 4.2.2    | Problem formulation . . . . .                                         | 85        |
| 4.2.3    | Recursive estimation of the electrical submodel . . . . .             | 86        |
| <b>5</b> | <b>Conclusions and future work</b>                                    | <b>90</b> |
| 5.1      | Conclusions . . . . .                                                 | 90        |
| 5.2      | Future work . . . . .                                                 | 91        |
| <b>A</b> | <b>Digital implementation of CT filtering operations</b>              | <b>92</b> |
| A.1      | Zero-order-hold assumption . . . . .                                  | 92        |
| A.2      | Trapezoidal approximation . . . . .                                   | 93        |
| <b>B</b> | <b>Stability of LTV systems</b>                                       | <b>94</b> |
| <b>C</b> | <b>Scaled version of Potter's algorithm</b>                           | <b>95</b> |



# List of Abbreviations

|          |                                                                                               |
|----------|-----------------------------------------------------------------------------------------------|
| ARX      | AutoRegressive with eXogenous inputs                                                          |
| CLRIVSVF | Closed-Loop Recursive Instrumental Variable State-Variable Filter                             |
| CLRSRIVC | Closed-Loop Recursive Simplified Refined Instrumental Variable for<br>Continuous-time systems |
| CT       | Continuous-Time                                                                               |
| DT       | Discrete-Time                                                                                 |
| IV       | Instrumental Variable                                                                         |
| KF       | Kalman Filter                                                                                 |
| LMR      | Levenberg-Marquardt Regularization                                                            |
| LTI      | Linear Time-Invariant                                                                         |
| LTV      | Linear Time-Varying                                                                           |
| OE       | Output Error                                                                                  |
| PEM      | Prediction-Error Method                                                                       |
| RIVSVF   | Recursive Instrumental Variable State-Variable Filter                                         |
| RLSSVF   | Recursive Least Squares State-Variable Filter                                                 |
| RSRIVC   | Recursive Simplified Refined Instrumental Variable for Continuous-time<br>systems             |
| SISO     | Single-Input Single-Output                                                                    |
| SNR      | Signal-to-Noise Ratio                                                                         |
| SRIVC    | Simplified Refined Instrumental Variable method for Continuous-time<br>systems                |
| SVF      | State-Variable Filter                                                                         |

# List of Symbols

|                        |                                                   |
|------------------------|---------------------------------------------------|
| $F(p)$                 | prefilter                                         |
| $I_r$                  | identity matrix of order $r$                      |
| $p$                    | differentiation operator                          |
| $Q_n$                  | normalized covariance matrix of the Kalman filter |
| $q$                    | shift operator                                    |
| $t_k$                  | $k$ -th sampling instant                          |
| $u$                    | input signal                                      |
| $x$                    | noise-free output                                 |
| $y$                    | output signal                                     |
| $\mathcal{M}$          | model set                                         |
| $\mathcal{S}$          | true system                                       |
| $\ \cdot\ $            | euclidean norm                                    |
| $\zeta$                | instrument vector                                 |
| $\zeta_f$              | filtered instrument vector                        |
| $\theta$               | parameter vector                                  |
| $\theta_o$             | parameter vector of the true system               |
| $\kappa_p$             | $p$ -norm condition number                        |
| $\lambda_{\text{svf}}$ | cut-off frequency of the state-variable filter    |
| $\sigma^2$             | variance of the discrete-time noise $e$           |
| $\varphi$              | regressor vector                                  |
| $\varphi_f$            | filtered regressor vector                         |

# Chapter 1

## Introduction

People working in science and engineering are interested in understanding natural phenomena, and to this end they usually build models that represent the observed system. In the field of fluid dynamics for example, the interest is to understand how to modify the design of a ship in order to improve its maneuverability. If the ship is already built, the problem for a control engineer might be to design a controller to automatically guide the vessel. A mathematical model of the ship dynamics, consisting in a set of equations, would then serve those purposes [Fossen, 2002]. One approach to obtain a model is from physical principles like Newton's laws or Kirchhoff's laws. The second option for modeling is *system identification*, which basically consists in building a model based on collected measurements of the system input/output. This thesis focuses on the latter approach.

In the following, we first give some general ideas about system identification. Afterwards in Section 1.2, we present the indirect and direct approaches to identify continuous-time linear time-invariant models in an off-line fashion. Section 1.3 is dedicated to linear time-varying models. First we describe the type of linear time-varying models in which we are interested in. Putting the attention in recursive approaches, identification methods for the discrete-time and continuous-time cases are presented in Sections 1.3.2 and 1.3.3, respectively. Then, the indirect and direct identification approaches for linear time-varying systems are illustrated through a numerical example. Practical aspects and applications in recursive identification are presented in Sections 1.4 and 1.5, respectively. Finally, we present the organization of the thesis, contributions and publications.

### 1.1 System identification

As we already mentioned, in system identification the goal is to estimate a mathematical model of a given system (or process), based on observations of such system. The system identification procedure is shown in Figure 1.1, where we can see the three main entities, which are:

- Data set. It is a collection of measurements of the input  $u$  and output  $y$  of the system  $\mathcal{S}$ , and it is denoted by  $Z_N = \{u(t_k), y(t_k)\}_k^N$ , where  $N$  is the number of samples. Notice that  $y$  is contaminated by  $v$  which can represent a disturbance and/or measurement noise coming from the sensor.
- Model set. It is a set denoted by  $\mathcal{M}$  with candidate *model structures* that might represent the system [Ljung, 1999, p. 111]. A model structure is a mathematical expression, which relates  $u$ ,  $y$  and  $v$ , and it is parametrized by a vector  $\theta \in \mathbb{R}^{n_\theta}$ .
- Identification method. Given the first two entities, an identification method has

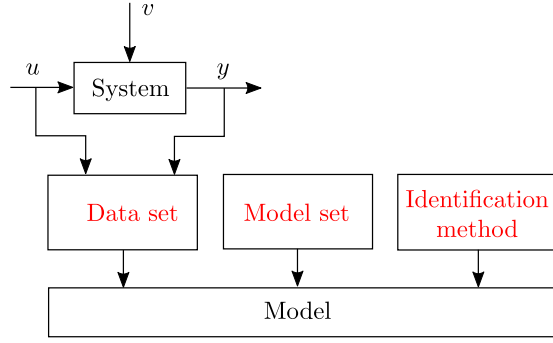


Figure 1.1: Basic entities in system identification: data set, model set and identification method.

to be chosen in order to estimate the parameter vector  $\theta$ , and as a consequence the model.

In practice there is always a difference between the true system  $\mathcal{S}$  and the model set  $\mathcal{M}$ , *i.e.* the system does not belong to the model set ( $\mathcal{S} \notin \mathcal{M}$ ). In other words, that means that there are unmodeled dynamics.

Models estimated using system identification techniques can be categorized as *gray*- or *black-box* models [Ljung, 1999]. In the former, physical principles like Newton's laws are used to build the model structure, which includes unknown parameters. Those parameters are then estimated considering a certain data set and identification method. On the other hand, in a black-box model, the structure and its parameters are solely determined by the data set and the identification method. Furthermore, the model can be in discrete-time (DT) or continuous-time (CT). In this work we are interested in CT linear models which can be of gray box or black box type.

## 1.2 Off-line identification of continuous-time models

In off-line estimation, also called non-recursive, batch or *en-bloc* identification, the whole data set must be available for estimation. In this section we illustrate with a simple example how to obtain a CT linear time-invariant model through a non-recursive identification approach.

### 1.2.1 Problem formulation

Many physical processes can be described by a CT output error (OE) model. This motivates the use of OE models in this study. As true system, let us consider then a single-input single output (SISO) CT OE representation with input  $u$  and output  $y$ , *i.e.*

$$\mathcal{S} \begin{cases} A_o(p)x(t) = B_o(p)u(t) \\ y(t_k) = x(t_k) + e_o(t_k) \end{cases} \quad (1.1)$$

where  $p$  is the differentiation operator;  $x(t)$  is the noise-free output; and the additive term  $e_o(t_k)$  is a zero-mean DT white noise sequence. The polynomials  $A_o(p)$  and  $B_o(p)$  are assumed to be relatively coprime and are given by

$$B_o(p) = b_0^o p^{n_b} + b_1^o p^{n_b-1} + \dots + b_{n_b}^o \quad (1.2)$$

$$A_o(p) = p^{n_a} + a_1^o p^{n_a-1} + \dots + a_{n_a}^o \quad (1.3)$$

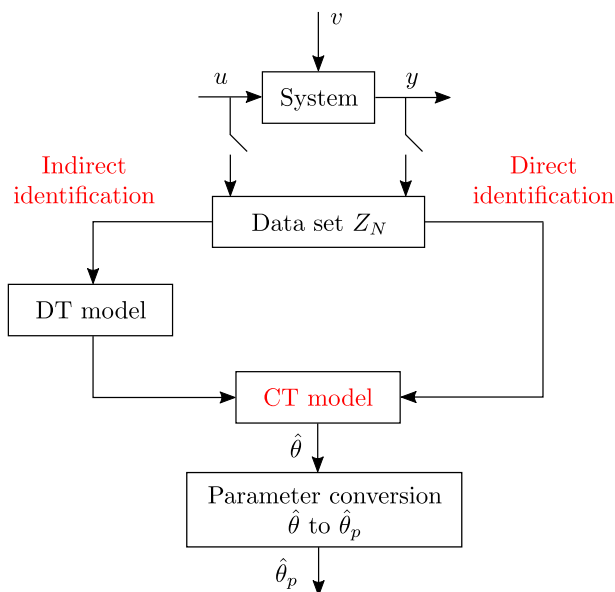


Figure 1.2: Indirect and direct routes to identification of a CT model.

where  $n_a \geq n_b$ . The argument  $t_k$  in the second equation in (1.1) denotes the sampled value of the associated variable at the  $k$ th sampling instant.

Measurements of the input and output with a sampling time  $T_s$  form the available data set  $Z_N = \{u(t_k); y(t_k)\}_{k=1}^N$ . Then, the goal is to propose a CT model structure parametrized with  $\theta$ , and to estimate  $\theta$  using the given data set and a certain identification method. We will assume that we know the system structure along with the polynomial degrees  $n_a$  and  $n_b$ , and the intersample behavior of the input.

### 1.2.2 The indirect vs direct identification methods

The two ways in which a CT model can be obtained are illustrated in the scheme of Figure 1.2; they are [Garnier et al., 2008]:

- Indirect approach. First, a DT model is identified and then it is converted to CT. In the LTI case, this can be done in MATLAB using the routine `d2c`.
- Direct approach. It aims at estimating the CT model directly.

As shown in the scheme, the CT parameters can be converted to physical parameters  $\hat{\theta}_p$ , if necessary. Notice that in both the indirect and direct approaches, we need to assume a certain intersample behavior [Garnier et al., 2008].

Motivated by the advantages that offers the direct approach over the indirect approach (see [Garnier et al., 2008, Garnier and Young, 2014]), we have decided to work in this study with the former approach. Some of the advantages of the direct approach are the following:

- It allows us to estimate directly physical quantities.
- It is suitable for non-uniformly sampled data.
- It avoids the transformation between DT and CT models which can be difficult.
- It is not sensitive to high sampling rates.

Different direct approaches are available and they can be classified as follows [Garnier et al., 2003, 2008]:

- linear filter methods
- integral methods
- modulating function methods
- other approaches like finite difference methods

In this study we consider linear filter methods.

### Indirect identification using a DT ARX model

A simple representation for the true system  $\mathcal{S}$  given in (1.1) is the DT autoregressive with exogenous input (ARX) model [Ljung, 1999]. By assuming that we know the system structure, the order of the system and the intersample behavior of the input, we can define the corresponding DT ARX representation which is given by

$$\mathcal{M}: \quad y(t_k) + \alpha_1 y(t_{k-1}) + \dots + \alpha_{n_\alpha} y(t_{k-n_\alpha}) = \beta_0 u(t_k) + \dots + \beta_{n_\beta} u(t_{k-n_\beta}) + e(t_k) \quad (1.4)$$

where  $e(t_k)$  is assumed to be a DT white noise sequence. Model (1.4) can be written as the linear regression

$$y(t_k) = \phi^T(t_k) \rho + e(t_k) \quad (1.5)$$

with the regressor

$$\phi(t_k) = \begin{bmatrix} -y(t_{k-1}) & \dots & -y(t_{k-n_\alpha}) & u(t_k) & \dots & u(t_{k-n_\beta}) \end{bmatrix}^T \quad (1.6)$$

and the parameter vector  $\rho \in \mathbb{R}^{n_\rho}$ , *i.e.*

$$\rho = \begin{bmatrix} \alpha_1 & \dots & \alpha_{n_\alpha} & \beta_0 & \dots & \beta_{n_\beta} \end{bmatrix}^T \quad (1.7)$$

The corresponding predictor is given by

$$\hat{y}(t_k|\rho) = \phi^T(t_k) \rho \quad (1.8)$$

The parameter vector  $\rho$  can be obtained by minimizing the prediction error

$$\begin{aligned} \varepsilon(t_k) &= y(t_k) - \hat{y}(t_k|\rho) \\ &= y(t_k) - \phi^T(t_k) \rho \end{aligned} \quad (1.9)$$

*i.e.*

$$\hat{\rho} = \arg \min_{\rho \in \mathbb{R}^{n_\rho}} V \quad (1.10)$$

Usually,  $\rho$  is computed by choosing  $V$  as the mean-square error (MSE) defined by

$$V = \frac{1}{N} \sum_{k=1}^N [y(t_k) - \phi^T(t_k) \rho]^2 \quad (1.11)$$

This approach is the well-known prediction error method (PEM). The advantage of the DT ARX model is that (1.10) leads to the normal equations

$$\left[ \sum_{k=1}^N \phi(t_k) \phi^T(t_k) \right] \hat{\rho} = \left[ \sum_{k=1}^N \phi(t_k) y(t_k) \right] \quad (1.12)$$

which have the closed-form solution

$$\hat{\rho} = \left[ \sum_{k=1}^N \phi(t_k) \phi^T(t_k) \right]^{-1} \cdot \left[ \sum_{k=1}^N \phi(t_k) y(t_k) \right] \quad (1.13)$$

Equation (1.13) corresponds to the least squares estimates.

### Indirect identification using a DT OE model

By assuming that we know the system structure, the order of the system and the inter-sample behavior of the input, we can define the corresponding DT OE representation which is given by

$$\mathcal{M} \begin{cases} x(t_k) + \alpha_1 x(t_{k-1}) + \dots + \alpha_{n_\alpha} x(t_{k-n_\alpha}) = \beta_0 u(t_k) + \dots + \beta_{n_\beta} u(t_{k-n_\beta}) \\ y(t_k) = x(t_k) + e(t_k) \end{cases} \quad (1.14)$$

with  $e(t_k)$  a zero-mean DT white noise sequence. The predictor for (1.14) is defined as follows [Ljung, 1999, p. 85]

$$\hat{y}(t_k|\rho) = \phi^T(t_k, \rho)\rho \quad (1.15)$$

with the parameter vector  $\rho$  defined in (1.7), and the regressor

$$\phi(t_k, \rho) = \begin{bmatrix} -\hat{x}(t_{k-1}, \rho) & \dots & \hat{x}(t_{k-n_\alpha}, \rho) & u(t_k) & \dots & u(t_{k-n_\beta}) \end{bmatrix}^T \quad (1.16)$$

where  $\hat{x}(t_k, \rho)$  is obtained as the solution of the difference equation

$$\hat{x}(t_k, \rho) + \alpha_1 \hat{x}(t_{k-1}, \rho) + \dots + \alpha_{n_\alpha} \hat{x}(t_{k-n_\alpha}, \rho) = \beta_0 u(t_k) + \dots + \beta_{n_\beta} u(t_{k-n_\beta}) \quad (1.17)$$

Note that (1.15) is non-linear due to the dependence of  $\rho$  in the regressor vector. To make the difference between (1.8) and (1.15), the latter is called pseudolinear regression.

As before, the parameters can be obtained by minimizing the prediction error. However, due to the non-linear predictor (1.15), (1.10) corresponds to a non-convex optimization problem, and it cannot be guaranteed that the numerical search will yield to a global minimum. An alternative to solve the problem is to use instrumental variable techniques as for example the simplified refined instrumental variable method (SRIV) [Young, 2011]. In that approach, the non-convex optimization is replaced by iterations of convex optimization problems.

### Direct identification of a CT ARX model

We assume here that the system can be represented by a CT ARX model, *i.e.*

$$\mathcal{M} : \quad y^{(n_a)}(t_k) + a_1 y^{(n_a-1)}(t_k) + \dots + a_{n_a} y(t_k) = b_0 u^{(n_b)}(t_k) + \dots + b_{n_b} u(t_k) + e(t_k) \quad (1.18)$$

where  $x^{(i)}(t_k)$  denotes the  $i$ th time derivative of the continuous-time signal  $x(t)$  at time instant  $t_k = kT_s$ , and  $e(t_k)$  is a zero-mean DT white noise.

The model (1.18) cannot be directly identified because the time-derivatives are usually not available. One standard approach to solve this problem is to use linear filter methods, such as the state-variable filter (SVF) (see *e.g.* [Garnier et al., 2008] and the prior references therein). The approach consists in filtering the model with a SVF, which is a low-pass CT filter defined by <sup>1</sup>

$$F(p) = \frac{1}{(p + \lambda_{\text{svf}})^{n_a}} \quad (1.19)$$

Then, the filtered signals  $y_f^{(i)}(t_k)$  and  $u_f^{(i)}(t_k)$  are obtained. They correspond to the prefiltered time-derivatives of the input and output in the bandwidth of interest, defined via the choice of  $\lambda_{\text{svf}}$ . This can be written as

$$y_f^{(i)}(t_k) = F(p)y^{(i)}(t_k) \quad i = 0, \dots, n_a \quad (1.20a)$$

$$u_f^{(i)}(t_k) = F(p)u^{(i)}(t_k) \quad i = 0, \dots, n_b \quad (1.20b)$$

<sup>1</sup>The static gain can be made unity.

or equivalently as

$$y_f^{(i)}(t_k) = p^i F(p)y(t_k) \quad i = 0, \dots, n_a \quad (1.21a)$$

$$u_f^{(i)}(t_k) = p^i F(p)u(t_k) \quad i = 0, \dots, n_b \quad (1.21b)$$

Applying  $F(p)$  to (1.18) we obtain the following equation (which is nearly exact, except for transients arising from initial conditions),

$$y_f^{(n_a)}(t_k) + a_1 y_f^{(n_a-1)}(t_k) + \dots + a_{n_a} y_f(t_k) = b_0 u_f^{(n_b)}(t_k) + \dots + b_{n_b} u_f(t_k) + v_f(t_k) \quad (1.22)$$

with

$$v_f(t_k) = F(p)e(t_k) \quad (1.23)$$

Equation (1.22) can be rewritten as a linear regression,

$$y_f^{(n_a)}(t_k) = \varphi_f^T(t_k)\theta + v_f(t_k) \quad (1.24)$$

where

$$y_f^{(n_a)}(t_k) = p^{n_a} F(p)y(t_k) \quad (1.25a)$$

$$\varphi_f^T(t_k) = F(p)\varphi^T(t_k) \quad (1.25b)$$

with the regression vector

$$\varphi^T(t_k) = [-y^{(n_a-1)}(t_k) \quad \dots \quad -y(t_k) \quad u^{(n_b)}(t_k) \quad \dots \quad u(t_k)] \quad (1.26)$$

*i.e.* its filtered version is given by

$$\varphi_f^T(t_k) = [-y_f^{(n_a-1)}(t_k) \quad \dots \quad -y_f(t_k) \quad u_f^{(n_b)}(t_k) \quad \dots \quad u_f(t_k)] \quad (1.27)$$

The parameter vector  $\theta \in \mathbb{R}^{n_\theta}$  is defined by

$$\theta = [a_1 \quad \dots \quad a_{n_a} \quad b_0 \quad \dots \quad b_{n_b}]^T \quad (1.28)$$

The predictor for (1.18) is then given by

$$\hat{y}_f^{(n_a)}(t_k|\theta) = \varphi_f^T(t_k)\theta \quad (1.29)$$

The parameter vector (1.28) can be estimated by the least squares method, *i.e.* by solving the minimizing problem

$$\hat{\theta} = \arg \min_{\theta \in \mathbb{R}^{n_\theta}} V \quad (1.30)$$

with

$$V = \frac{1}{N} \sum_{k=1}^N [y_f^{(n_a)}(t_k) - \varphi_f^T(t_k)\theta]^2 \quad (1.31)$$

The closed form solution to (1.30) is

$$\hat{\theta} = \left[ \sum_{k=1}^N \varphi_f(t_k)\varphi_f^T(t_k) \right]^{-1} \cdot \left[ \sum_{k=1}^N \varphi_f(t_k)y_f^{(n_a)}(t_k) \right] \quad (1.32)$$

This estimation method is called the least squares state-variable filter (LSSVF) approach.

### Direct identification of a CT OE model

We assume here that the system can be represented by a CT OE model, *i.e.*

$$\mathcal{M} \begin{cases} x^{(n_a)}(t_k) + a_1 x^{(n_a-1)}(t_k) + \dots + a_{n_a} x(t_k) = b_0 u^{(n_b)}(t_k) + \dots + b_{n_b} u(t_k) \\ y(t_k) = x(t_k) + e(t_k) \end{cases} \quad (1.33)$$

Analogously to the DT case, the predictor  $y(t_k|\theta)$  corresponds to a pseudolinear regression. However, in this case it involves the solution of the differential equation

$$\hat{x}^{(n_a)}(t_k, \theta) + a_1 \hat{x}^{(n_a-1)}(t_k, \theta) + \dots + a_{n_a} \hat{x}(t_k, \theta) = b_0 u^{(n_b)}(t_k) + \dots + b_{n_b} u(t_k) \quad (1.34)$$

Then, the PEM estimate given by

$$\hat{\theta} = \arg \min_{\theta \in \mathbb{R}^{n_\theta}} \frac{1}{N} \sum_{k=1}^N [y(t_k) - \hat{y}(t_k|\theta)]^2 \quad (1.35)$$

is again a non-convex optimization problem. Alternatives to avoid the non-convex optimization are the suboptimal instrumental variable state-variable filter (IVSVF) method and the simplified refined instrumental variable method for CT models (SRIVC) [Young et al., 2008], which is optimal for CT OE models. In Chapter 2, both IVSVF and SRIVC will be recalled and then adapted to linear time-varying systems.

## 1.3 Identification of linear time-varying models

The purpose of this section is to show how to extend the direct approach for the estimation of CT models to the case of linear time-varying (LTV) systems. The identification of CT LTV models shares methodological aspects with the DT LTV case. Therefore, after some comments about LTV systems, we discuss approaches that have been proposed in DT LTV identification.

### 1.3.1 Linear time-varying systems

A time-varying process can be defined as a system whose properties vary in time. In real-life, processes are usually time-varying, for example, due to aging effects or changes in the environmental conditions such as the temperature. Sometimes the time scale in which we are interested in, is short enough, to consider the system as time-invariant. However, this is not the case for the systems that we deal with in this thesis.

The parameters of a time-varying process can vary in the following four ways [Niedźwiecki, 2000, p. 60]:

- Slow persistent changes (sometimes called parameter drift)
- Infrequent abrupt changes
- Mixed-mode variations (slow and abrupt changes)
- All other changes (also called fast parameter variations)

Note, that there is no precise distinction between fast and slow parameter variation, although several attempts to quantify the degree of non-stationarity have been done [Niedźwiecki, 2000, p. 67]. In this work, it is assumed that the parameters change slowly in time. In addition we restrain our attention to linear processes, that is, we will focus on slowly linear time-varying (LTV) systems.

LTV systems can be considered as a special class of linear parameter-varying (LPV) systems. Contrary to an LTV system, the dynamics of an LPV process depend on an independent external signal called scheduling variable, that can be measured. A typical example of an LPV process is an aircraft, where the scheduling parameters are the speed and altitude. Notice that for a given trajectory of the scheduling variable, an LTV system is obtained. Due to the similarities between these type of systems, techniques developed for LPV systems could be extended to LTV systems and vice versa. Identification of LPV models has been recently addressed in *e.g.* [Tóth, 2010, Laurain et al., 2011].

### 1.3.2 Identification of discrete-time models

#### Non-recursive identification methods using a sliding window

Batch estimation can also be used for the estimation of slowly LTV systems. A simple and intuitive approach is to estimate local LTI models through the weighted least squares in a sliding window. Two possible options are the following [Niedźwiecki, 2000]:

- Sliding window least squares, which considers a rectangular window, meaning that all the weights are equal.
- Exponentially weighted least squares, where old data is discarded by means of exponential weights.

The identification of time-varying systems can also be tackled using deterministic models for the parameter variations, and then performing the estimation with the batch sliding window approach. Such a technique is explored for instance in [Chan and Zhang, 2011] by means of local polynomial modeling. Nevertheless, we are interested in on-line identification, where it is hard to find an appropriate representation of the time-varying parameters using deterministic models. Additionally the number of parameters is increased, leading to estimates with larger variances.

#### Recursive identification methods

In recursive estimation, also called on-line or real-time identification, the model is sequentially updated every time new measurements become available. In comparison with batch estimation, recursive algorithms are in general advantageous in terms of having less computational cost. In this thesis, we focus on recursive identification.

**Remark 1.1** (*Recursive on-line and recursive off-line identification*). Note that recursive algorithms can be applied in an off-line or on-line fashion. In the former case, iterations can be used to improve the estimates (see *e.g.* [Solbrand et al., 1985, Young, 2015]). Nevertheless, recursive identification is particularly suitable for on-line applications in which the model parameters are time-varying. Thus we can talk about recursive off-line or recursive on-line estimation, although in this study we concentrate on the latter. Here, whenever the word recursive is used alone, it would then mean recursive on-line.

A large number of studies have been conducted on the recursive estimation of DT models (see *e.g.* [Ljung and Söderström, 1983, Goodwin and Sin, 1984, Ljung and Gunnarsson, 1990, Ljung, 1999, Haykin, 2014] and the references therein). Some well-known algorithms, summarized in [Ljung, 1999], are:

- Recursive least squares method with forgetting factor (RLS-FF). This simple approach is derived from the exponentially weighted least squares. The drawback of

this approach is that it is not suitable if the parameters vary at different rates. In that case, a matrix forgetting factor could be used as it has been suggested in [Saelid and Foss, 1983] (see also [Niedźwiecki, 2000, p. 105]).

- Kalman filter (KF). This algorithm is obtained by assuming that the parameters vary as a random walk, which is a stochastic model. As the matrix forgetting factor based method, the advantage of KF is that it is suitable when the parameters vary at different speeds. The random walk is the simplest stochastic model from the *generalized random walk* class [Young, 2011, p. 94], which has enhanced flexibility (see also [Kitagawa and Gersch, 1984]).
- Recursive instrumental variable method (RIV). As the non-recursive counterpart, this algorithm is algebraically very similar to the RLS, but with the advantage that it has superior statistical properties. Different variants of RIV can be considered [Ljung and Söderström, 1983]. To track time-varying parameter variations, RIV can be used together with a forgetting factor or the KF as adaptation mechanism [Young, 2011].
- Recursive prediction error method (RPEM). This algorithm is a gradient based approach. The RLS and the least mean squares (LMS) method are part of this family. Note that LMS and RLS are equally good for tracking slowly varying parameters [Lindbom, 1995, p. 16].

Recursive estimation can be also performed using deterministic models. As we already mentioned, in on-line identification, it is hard to find an appropriate representation of the time-varying parameters using deterministic models. Nevertheless, deterministic modeling can offer a good performance for the problem of fast parameter variations [Niedźwiecki, 2000]. Such a problem is addressed for instance in [Niedźwiecki and Klaput, 2002] using more than one deterministic model in a parallel estimation scheme; in this way, the difficulty of finding a deterministic model is circumvented. Deterministic models are also considered in the case of certain models for nonlinear systems, such as *state-dependent parameter* (see *e.g.* [Young et al., 2001, Young, 2011]) and *nonlinear parameter varying* (see *e.g.* [Previdi and Lovera, 2004]) models, although in this case they are used to characterize the relationship between the parameter variations and the measured system variables on which they are dependent. Finally, it is also valid to combine deterministic and stochastic models, as in the case of *dynamic harmonic regression* modeling [Young et al., 1999, Young, 2011].

Research in recursive estimation of DT time-varying models has continued in the last decades, for instance, in the estimation of models with fast parameter variations. To deal with that, methods based on the generalized random walk model class have been developed. Such approaches are presented in [Zheng and Lin, 2003] considering a RLS based algorithm and in [Rutström, 2005] considering a LMS based algorithm. Another recent topic of research is the estimation of non-parametric models in the time-domain, *i.e.* FIR models, using a regularization/Bayesian approach (see [Prando et al., 2016] and references therein).

The amount of recursive identification methods is extensive. Therefore, choosing an appropriate method is not an easy task. Criteria to make a good choice are for instance rate of convergence, tracking ability, robustness, computational requirements and numerical properties (see more in [Haykin, 2014, p. 23]). For the reasons given above, in this work we will propose algorithms based on RLS, KF and RIV.

### General recursive identification problem

Usually, the considered models can be written as a [Ljung, 1999]:

- Linear regression. For instance, the ARX model and finite impulse response (FIR) model belong to this class. The corresponding predictor is given by

$$\hat{y}(t_k|\rho) = \phi^T(t_k)\hat{\rho}(t_{k-1}) \quad (1.36)$$

- Pseudolinear regression. For instance, the autoregressive moving average with exogenous input (ARMAX) model and the OE model belong to this class. The corresponding predictor is given by

$$\hat{y}(t_k|\rho) = \phi^T(t_k, \rho)\hat{\rho}(t_{k-1}) \quad (1.37)$$

In Section 1.2.2, the predictors for ARX and OE models were defined. The only difference here is that the parameter vector can vary in time.

The models can be identified using recursive algorithms which have usually the following general form

$$\hat{\rho}(t_k) = \hat{\rho}(t_{k-1}) + \mu(t_k)L(t_k)\varepsilon(t_k) \quad (1.38)$$

where  $\hat{\rho}(t_k)$  is the estimated time-varying parameter vector and  $\varepsilon(t_k)$  the prediction error given by

$$\varepsilon(t_k) = y(t_k) - \hat{y}(t_k|\rho) \quad (1.9)$$

with  $\hat{y}(t_k|\rho)$  defined in (1.36) or (1.37) depending on the chosen model. The vector  $L(t_k)$  is the adaptation gain, and the scalar  $\mu(t_k)$  the step size; both  $L(t_k)$  and  $\mu(t_k)$  depend on the chosen algorithm [Ljung and Gunnarsson, 1990, Ljung, 1999]. In this thesis, we consider algorithms that belong to (1.38) with  $\mu(t_k) = 1$ . Next we review two of them, the RLS and KF methods.

### Recursive least squares method

The RLS method is derived from the exponentially weighted least squares, *i.e.* from the following optimization problem [Ljung, 1999]

$$\hat{\rho} = \arg \min_{\rho \in \mathbb{R}^{n_\rho}} \sum_{k=1}^{N'} \beta(N', k) [y(t_k) - \hat{y}(t_k|\rho)]^2 \quad (1.39)$$

where  $\hat{y}(t_k|\rho)$  is the predictor (1.8), which corresponds to the linear regression (1.5).  $\beta(N', k)$  is defined in terms of the forgetting factor  $\lambda(t_j)$ ,

$$\beta(N', k) = \begin{cases} \prod_{j=k+1}^{N'} \lambda(t_j) & 1 \leq k \leq N' - 1 \\ 1 & k = N' \end{cases} \quad (1.40)$$

Here we consider a constant forgetting factor  $\lambda$ . From (1.39) it can be shown that the recursive least squares with forgetting factor method (RLS-FF) is given by [Ljung and Söderström, 1983, Ljung, 1999]

$$\hat{\rho}(t_k) = \hat{\rho}(t_{k-1}) + L(t_k)\varepsilon(t_k) \quad (1.41a)$$

$$\varepsilon(t_k) = y(t_k) - \phi^T(t_k)\hat{\rho}(t_{k-1}) \quad (1.41b)$$

$$L(t_k) = \frac{P(t_{k-1})\phi(t_k)}{\lambda + \phi^T(t_k)P(t_{k-1})\phi(t_k)} \quad (1.41c)$$

$$P(t_k) = \frac{1}{\lambda} [P(t_{k-1}) - L(t_k)\phi^T(t_k)P(t_{k-1})] \quad (1.41d)$$

To start RLS, the initial conditions  $\hat{\theta}(t_0)$  and  $P(t_0)$  have to be specified. Usually, these initial conditions are set to  $\hat{\theta}(t_0) = 0$  and  $P(t_0) = c \cdot I_{n_\rho}$ , where  $c$  is a constant and  $I$  is the identity matrix of order  $n_\rho$  [Ljung and Söderström, 1983, p. 21].

### Kalman filter method

Let us assume that the model is the linear regression (1.5), and that the parameter variations can be modeled as a random walk model; therefore the full model becomes

$$\mathcal{M} \begin{cases} \rho(t_k) = \rho(t_{k-1}) + w(t_k) \\ y(t_k) = \phi^T(t_k)\rho(t_k) + e(t_k) \end{cases} \quad (1.42)$$

where  $w(t_k)$  and  $e(t_k)$  are zero-mean Gaussian noises with covariance matrix  $Q_w$  and variance  $\sigma_e^2$ , respectively. An estimate of  $\rho(t_k)$  can be obtained from a bayesian perspective, by further assuming that  $\rho(t_k)$  has a Gaussian distribution with mean  $\rho_0$  and covariance matrix  $P_0^*$ . The KF gives then the mean  $\hat{\rho}(t_k)$  and covariance matrix  $P^*(t_k)$  of the posterior distribution [Ljung and Söderström, 1983]. The KF algorithm is given by (see *e.g.* [Young, 2011])

*Prediction step:*

$$\hat{\rho}(t_k|t_{k-1}) = \hat{\rho}(t_{k-1}) \quad (1.43a)$$

$$P^*(t_k|t_{k-1}) = P^*(t_{k-1}) + Q_w \quad (1.43b)$$

*Correction step:*

$$\hat{\rho}(t_k) = \hat{\rho}(t_k|t_{k-1}) + L(t_k)\varepsilon(t_k) \quad (1.43c)$$

$$\varepsilon(t_k) = y(t_k) - \phi^T(t_k)\hat{\rho}(t_k|t_{k-1}) \quad (1.43d)$$

$$L(t_k) = \frac{P^*(t_k|t_{k-1})\phi(t_k)}{\sigma_e^2 + \phi^T(t_k)P^*(t_k|t_{k-1})\phi(t_k)} \quad (1.43e)$$

$$P^*(t_k) = P^*(t_k|t_{k-1}) - L(t_k)\phi^T(t_k)P^*(t_k|t_{k-1}) \quad (1.43f)$$

The normalized KF, obtained from (1.43) by considering that

$$P^*(t_k)/\sigma_e^2 = P(t_k) \quad (1.44)$$

is given by

*Prediction step:*

$$\hat{\rho}(t_k|t_{k-1}) = \hat{\rho}(t_{k-1}) \quad (1.45a)$$

$$P(t_k|t_{k-1}) = P(t_{k-1}) + Q_n \quad (1.45b)$$

*Correction step:*

$$\hat{\rho}(t_k) = \hat{\rho}(t_k|t_{k-1}) + L(t_k)\varepsilon(t_k) \quad (1.45c)$$

$$\varepsilon(t_k) = y(t_k) - \phi^T(t_k)\hat{\rho}(t_k|t_{k-1}) \quad (1.45d)$$

$$L(t_k) = \frac{P(t_k|t_{k-1})\phi(t_k)}{1 + \phi^T(t_k)P(t_k|t_{k-1})\phi(t_k)} \quad (1.45e)$$

$$P(t_k) = P(t_k|t_{k-1}) - L(t_k)\phi^T(t_k)P(t_k|t_{k-1}) \quad (1.45f)$$

where  $Q_n$  is the normalized covariance matrix, also called *noise-variance ratio* matrix, and is given by

$$Q_n = \frac{Q_w}{\sigma_e^2} \quad (1.46)$$

**Remark 1.2** *Let us consider that the true system is defined by a random walk and a linear regression, i.e.*

$$\mathcal{S} \begin{cases} \rho^o(t_k) = \rho^o(t_{k-1}) + w_o(t_k) \\ y(t_k) = \phi^T(t_k) \rho^o(t_k) + e_o(t_k) \end{cases} \quad (1.47)$$

where  $w_o(t_k)$  and  $e_o(t_k)$  are zero-mean Gaussian noises with covariance matrix  $Q_{w_o}$  and variance  $\sigma_{e_o}^2$ , respectively. Then, if the system belongs to the model set ( $\mathcal{S} \in \mathcal{M}$ ), the KF estimate  $\hat{\rho}(t_k)$  is optimal, in the sense that it minimizes the a posteriori parameter error covariance matrix, if KF is run with  $Q_{w_o}$  and  $\sigma_{e_o}^2$ . However, for other models, the KF is an ad hoc algorithm [Ljung and Gunnarsson, 1990, Ljung, 1999].

**Remark 1.3** *The matrix  $Q_w$  is usually set as*

$$Q_w = \text{diag}\{\sigma_{w_1}^2 \dots \sigma_{w_{n_\rho}}^2\} \quad (1.48)$$

where  $\sigma_{w_i}^2$  is the expected variation of the parameter  $\rho_i(t_k)$ . If all the parameters vary at the same speed, then  $Q_w = \sigma_w^2 I_{n_\rho}$ . On the other hand, if the parameter  $\rho_i(t_k)$  is known to be constant, then  $\sigma_{w_i}^2$  can be set to zero or a value close to zero.

### 1.3.3 Identification of continuous-time models

A recent study concerning identification of both DT and CT LTV models is for instance [Lataire, 2011], where an off-line frequency domain method has been developed. Subspace techniques applied in a recursive fashion have been also used to identify multiple-input multiple-output (MIMO) systems [Bergamasco and Lovera, 2011].

However, the DT identification methods presented above could be applied to identify a CT model using the indirect approach. Indirect recursive estimation has been used for example in [Gustafsson and Olsson, 1999], although this is not a common approach, because the advantages of the direct approach for LTI systems are also valid in the LTV case.

Regarding direct approaches, one option is to use integral methods (see [Garnier et al., 2003]), like block-pulse functions that are used in [Jiang and Schaufelberger, 1991]. The CT RPEM presented in [Östring and Gunnarsson, 2004] is also an integral method, and it has been applied to obtain physical parameters of a flexible robot arm. Another class of direct CT identification methods is the modulating function. A batch scheme with a recursive implementation using modulating functions is proposed in [Co and Ungarala, 1997, Ungarala and Co, 2000] for on-line estimation. A third class of direct CT identification approaches is the linear filter methods. For instance, an on-line linear filter based approach is presented in [Dimogianopoulos and Lozano, 2001], where a batch estimation using the exponentially weighted least squares is considered. Recursive algorithms, like RLS, RIV and the Kalman filter, can be coupled with linear filter methods [Åström and Wittenmark, 2008, p. 59]. That is done in [Isermann and Münchhof, 2010], where the recursive version of LSSVF, presented in Section 1.2.2, is used to track not only slowly but also abrupt parameter variations (see also [Canudas de Wit, 1986]). In this study we focus on linear filter based recursive algorithms but only for CT systems whose parameters vary slowly in time.

#### Indirect vs direct identification

Analogously to the batch estimation case, recursive identification can be used to identify CT LTV models either in a indirect or direct way. To illustrate the two approaches, let

us consider a CT LTV OE system,

$$\mathcal{S} \begin{cases} A_o(p, t)x(t) = B_o(p, t)u(t) \\ y(t_k) = x(t_k) + e_o(t_k) \end{cases} \quad (1.49)$$

where  $e_o(t_k)$  is a zero-mean DT white noise sequence, and

$$B_o(p, t) = b_0^o(t)p^{n_b} + b_1^o(t)p^{n_b-1} + \dots + b_{n_b}^o(t) \quad (1.50)$$

$$A_o(p, t) = p^{n_a} + a_1^o(t)p^{n_a-1} + \dots + a_{n_a}^o(t) \quad (1.51)$$

In this Chapter, we assume that system  $\mathcal{S}$  can be represented by a CT LTV ARX model set<sup>2</sup>, *i.e.*

$$\mathcal{M}: \quad y^{(n_a)}(t_k) + a_1(t_k)y^{(n_a-1)}(t_k) + \dots + a_{n_a}(t_k)y(t_k) = b_0(t_k)u^{(n_b)}(t_k) + \dots + b_{n_b}(t_k)u(t_k) + e(t_k) \quad (1.52)$$

In the indirect approach, after building the predictor, the estimates are compute with (1.38). Since the system is LTV, the conversion from DT to CT has to be done in every recursion.

In the direct approach using linear filter methods, we need to write (1.52) as a linear regression. In the LTV case, this is not obvious since the differentiation operator  $p$  is not commutative with the time-dependent variables. Therefore, a filter cannot be applied in order to obtain prefiltered time-derivatives. However, if we assume that the parameters vary slowly, we can neglect the non-commutativity problem. This issue will be formalized and discussed further in Chapter 2. In the meantime, let us state that this is possible. Analogously to the LTI case (see (1.24)), applying a filter  $F(p)$  to (1.52) we obtain the filtered time-varying linear regression

$$y_f^{(n_a)}(t_k) = \varphi_f^T(t_k)\theta(t_k) + v_f(t_k) \quad (1.53)$$

with  $y_f^{(n_a)}(t_k)$  and  $\varphi_f(t_k)$  defined in (1.25), and  $v_f(t_k) = F(p)e(t_k)$ . The parameter vector is now

$$\theta(t_k) = \begin{bmatrix} a_1(t_k) & \dots & a_{n_a}(t_k) & b_0(t_k) & \dots & b_{n_b}(t_k) \end{bmatrix}^T \quad (1.54)$$

Then, the corresponding prediction error is

$$\varepsilon(t_k) = y_f^{(n_a)}(t_k) - \varphi_f^T(t_k)\hat{\theta}(t_{k-1}) \quad (1.55)$$

Once the predictor is obtained, the CT parameters are obtained with a recursive algorithm of the general class (1.38), *i.e.* in this case

$$\hat{\theta}(t_k) = \hat{\theta}(t_{k-1}) + \mu(t_k)L(t_k)\varepsilon(t_k) \quad (1.56)$$

A scheme of the indirect and direct recursive identification approaches, applied to a linear regression model, is presented in Figure 1.3. For a pseudolinear regression the procedure is quite similar, but more involved due to the computation of the predictor.

Note that the prefiltering is an inherent step of the linear filter methods. In DT recursive estimation, data filtering is not compulsory, but it is sometimes recommended to cope with disturbances [Åström and Wittenmark, 2008, p. 466] or to enhance a certain frequency band. As the  $p$  operator, the shift operator  $q$  is also not commutative with time-dependent variables. Thus, in DT recursive estimation, if the data is filtered, the non-commutativity problem mentioned above does also exist.

<sup>2</sup>A CT LTV OE model set will be considered in Chapter 2.

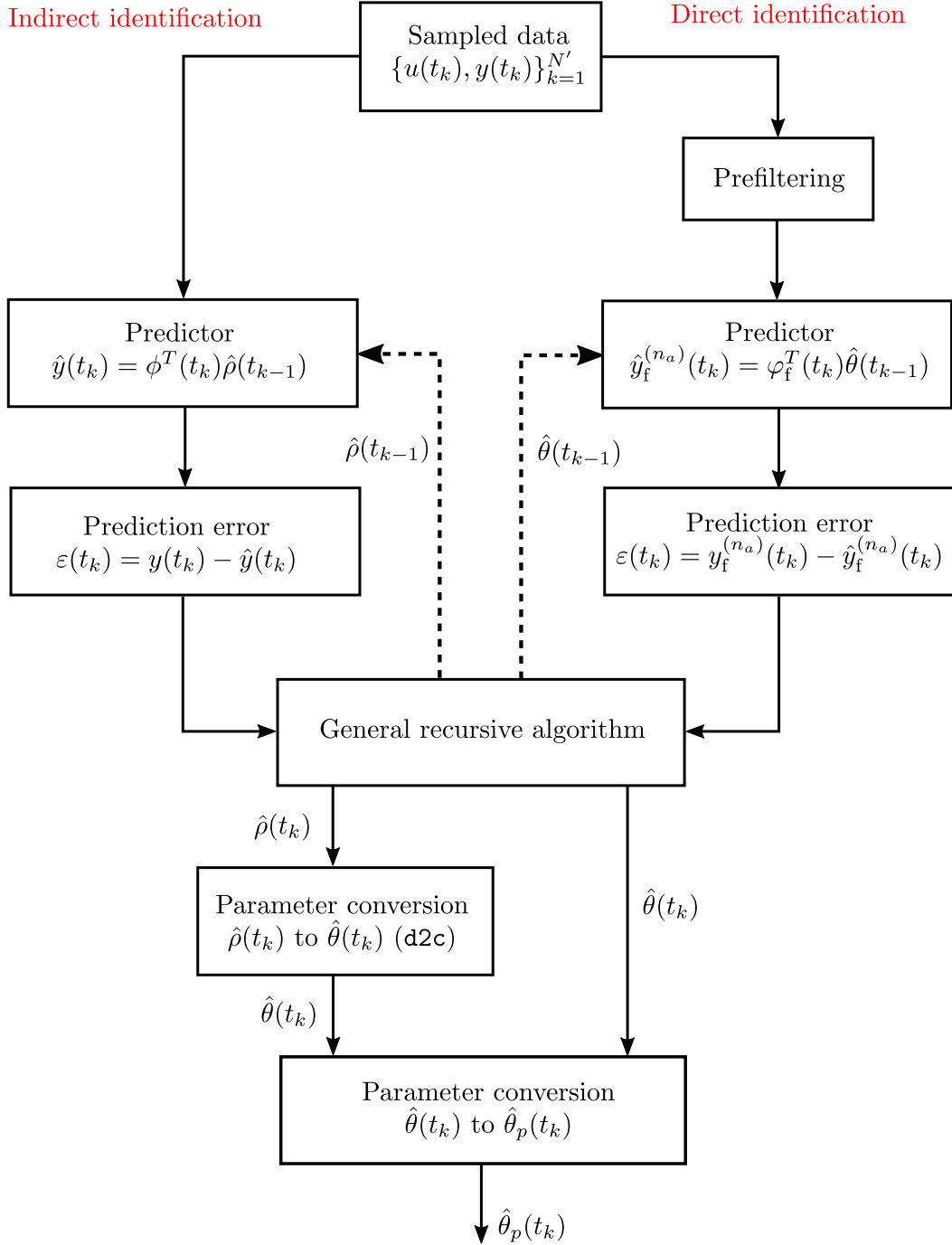


Figure 1.3: Indirect and direct recursive identification of a physical parameter vector  $\hat{\theta}_p(t_k)$ .

### Recursive least squares state-variable filter method

The direct LSSVF method presented in Section 1.2.2 can be implemented in a recursive fashion based on the methodology used in the DT case. To cope with the parameter variation we can consider the forgetting factor. Then the estimates are obtained with the following algorithm [Isermann and Münchhof, 2010]:

$$\hat{\theta}(t_k) = \hat{\theta}(t_{k-1}) + L(t_k)\varepsilon(t_k) \quad (1.57a)$$

$$\varepsilon(t_k) = y_f^{(na)}(t_k) - \varphi_f^T(t_k)\hat{\theta}(t_{k-1}) \quad (1.57b)$$

$$L(t_k) = \frac{P(t_{k-1})\varphi_f(t_k)}{\lambda + \varphi_f^T(t_k)P(t_{k-1})\varphi_f(t_k)} \quad (1.57c)$$

$$P(t_k) = \frac{1}{\lambda} \left[ P(t_{k-1}) - L(t_k)\varphi_f^T(t_k)P(t_{k-1}) \right] \quad (1.57d)$$

where  $y_f^{(na)}(t_k)$  and  $\varphi_f(t_k)$  are defined in (1.25). Algorithm (1.57) will be called the recursive least squares state-variable filter with forgetting factor (RLSSVF-FF) method.

The second option is to couple LSSVF with KF as an adaptation mechanism for the time-varying parameters. Then, the parameters are modeled through a random walk, *i.e.*

$$\theta(t_k) = \theta(t_{k-1}) + w(t_k) \quad (1.58)$$

The full model is given by (1.52) and (1.58), with  $w(t_k)$  and  $e(t_k)$  independent zero-mean DT Gaussian noise processes with covariance matrix  $Q_w$  variance  $\sigma_e^2$ , respectively.

Then, the algorithm, based on the DT methodology (see *e.g.* [Young, 2011]), is defined by

*Prediction step:*

$$\hat{\theta}(t_k|t_{k-1}) = \hat{\theta}(t_{k-1}) \quad (1.59a)$$

$$P(t_k|t_{k-1}) = P(t_{k-1}) + Q_n \quad (1.59b)$$

*Correction step:*

$$\hat{\theta}(t_k) = \hat{\theta}(t_k|t_{k-1}) + L(t_k)\varepsilon(t_k) \quad (1.59c)$$

$$\varepsilon(t_k) = y_f^{(na)}(t_k) - \varphi_f^T(t_k)\hat{\theta}(t_k|t_{k-1}) \quad (1.59d)$$

$$L(t_k) = \frac{P(t_k|t_{k-1})\varphi_f(t_k)}{1 + \varphi_f^T(t_k)P(t_k|t_{k-1})\varphi_f(t_k)} \quad (1.59e)$$

$$P(t_k) = P(t_k|t_{k-1}) - L(t_k)\varphi_f^T(t_k)P(t_k|t_{k-1}) \quad (1.59f)$$

where  $y_f^{(na)}(t_k)$  and  $\varphi_f(t_k)$  are defined in (1.25). Algorithm (1.59) will be called the recursive least squares state-variable filter based Kalman filter (RLSSVF-KF) method.

Two algorithms have been presented, RLSSVF-FF and RLSSVF-KF. In the thesis, we mainly use RLSSVF-KF; therefore, to simplify the notation we denote it just as RLSSVF.

To start RLSSVF, the initial conditions  $\hat{\theta}(t_0)$  and  $P(t_0)$  have to be specified. In this work, unless otherwise stated we choose  $\hat{\theta}(t_0) = 0$  and  $P(t_0) = 10^4 \cdot I_{n_\theta}$ .

**Remark 1.4** *The cut-off frequency  $\lambda_{\text{svf}}$  is a user parameter that should be chosen somewhat larger than the system bandwidth. In the LTV case and especially for systems with large variations in the bandwidth, the specification of  $\lambda_{\text{svf}}$  can be critical since the system bandwidth is time-varying. Some other aspects about the SVF and the choice of  $\lambda_{\text{svf}}$  are discussed in Section 3.1.*

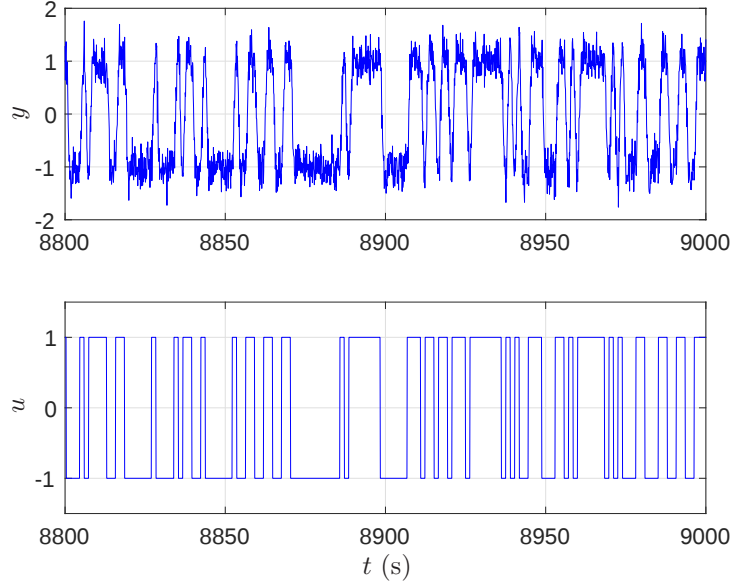


Figure 1.4: Part of the input-output data.

Notice that in CT recursive estimation, an additional computation is needed to obtain the simulated model output, which we can use to define the output error as follows

$$\varepsilon_y(t_k) = y(t_k) - \hat{y}(\hat{\theta}(t_k)) \quad (1.60)$$

#### 1.3.4 Numerical example - Similar parameter variations

We illustrate the indirect and direct approach with an example where the parameters vary at a similar rate. Let us consider the following CT LTV OE system

$$\mathcal{S} \begin{cases} (p^2 + a_1^o(t)p + a_2^o(t))x(t) = b_0^o(t)u(t) \\ y(t_k) = x(t_k) + e(t_k) \end{cases} \quad (1.61)$$

where

$$a_1^o(t) = 2\zeta_d\omega_n(t) \quad a_2^o(t) = b_0^o(t) = \omega_n^2(t) \quad (1.62)$$

with constant damping  $\zeta_d = 0.6$  and

$$\omega_n(t) = 2(1 + 0.3 \cos(2\pi t/4500)) \quad (1.63)$$

For this second order underdamped system, the measurements have a sampling time  $T_s = 0.1$  s and the total simulation time is 9000 s. Let us define the signal-to-noise ratio (SNR) as follows

$$\text{SNR} = 10 \log \left( \frac{P_x}{P_e} \right) \quad (1.64)$$

where  $P_x$  and  $P_e$  denote the average power of the noise-free output and the additive noise, respectively. In this example, the DT measurement noise is a zero-mean, Gaussian noise with constant variance 0.05, that corresponds to  $\text{SNR} \approx 11$  dB. The input of the system is a PRBS. Part of the input-output data is shown in Figure 1.4. The step responses and the Bode plots for some frozen systems obtained every 250 s are plotted in Figures 1.5 and 1.6, respectively. The bandwidth of the same frozen system are shown in Figure 1.7.

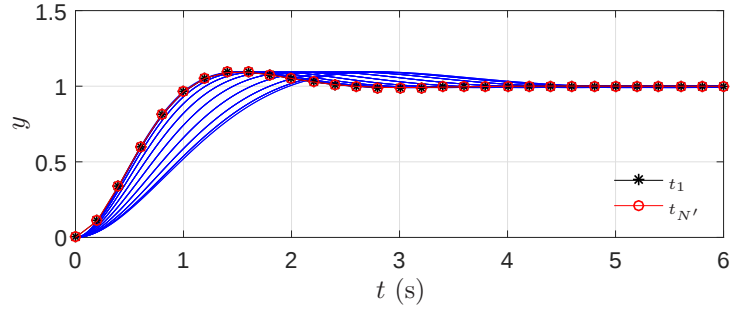


Figure 1.5: Step responses for some frozen systems obtained every 250 s. The system responses for  $t_1 = 0$  and  $t_{N'} = 9000$  s, matching in this case, are highlighted.

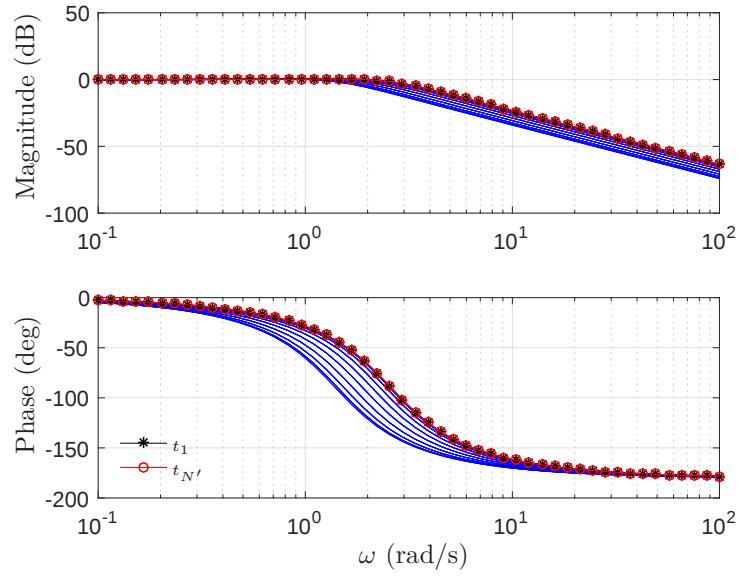


Figure 1.6: Bode plots for the frozen systems obtained every 250 s. The bode plots for  $t_1 = 0$  and  $t_{N'} = 9000$  s, matching in this case, are highlighted.

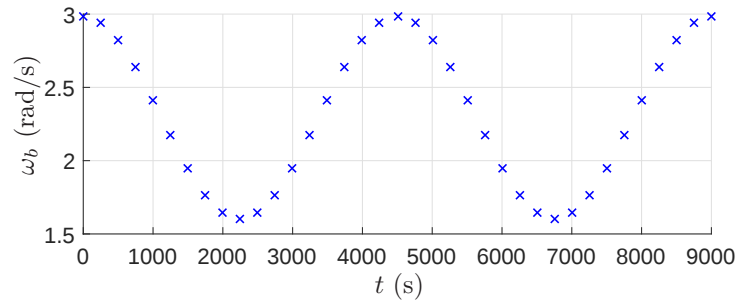


Figure 1.7: Bandwidth  $\omega_b$  for the frozen systems obtained every 250 s.

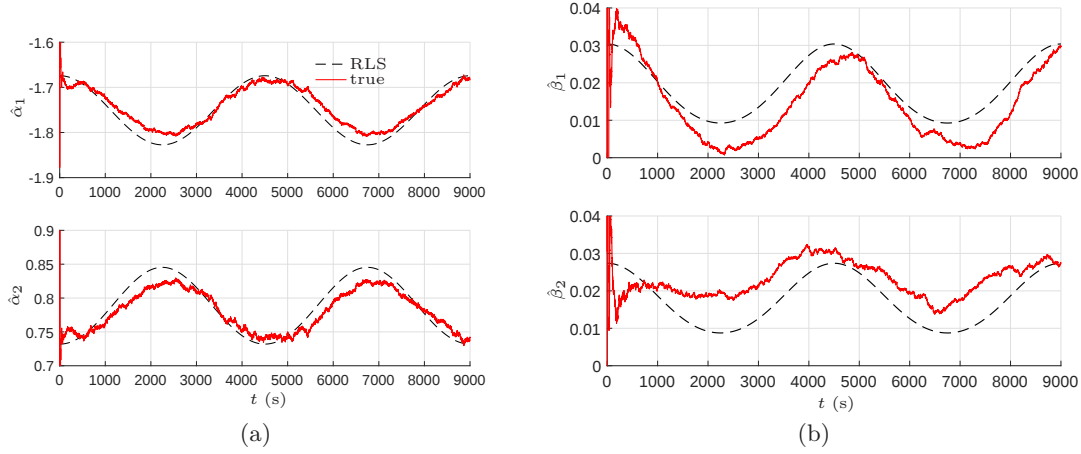


Figure 1.8: True DT parameters and RLS estimates of the DT model.

### Indirect vs direct identification

For the indirect identification we need to determine the discrete-time equivalent of the true system (1.61). Considering the appropriate zero-order hold assumption for the input, in this case, it is given by

$$x(t_k) + \alpha_1^o(t_k)x(t_{k-1}) + \alpha_2^o(t_k)x(t_{k-2}) = \beta_1^o(t_k)u(t_{k-1}) + \beta_2^o(t_k)u(t_{k-2}) \quad (1.65a)$$

$$y(t_k) = x(t_k) + e(t_k) \quad (1.65b)$$

For the model set we assume a DT LTV ARX representation, *i.e.*

$$y(t_k) + \alpha_1(t_k)x(t_{k-1}) + \alpha_2(t_k)y(t_{k-2}) = \beta_1(t_k)u(t_{k-1}) + \beta_2(t_k)u(t_{k-2}) + e(t_k) \quad (1.66)$$

The model is estimated with the MATLAB routine `recursiveARX` with a KF as adaptation mechanism. That corresponds to the KF algorithm (1.43). By trial and error, we define the normalized covariance matrix  $Q_n$  (see (1.46)), which is given by

$$Q_n = \text{diag}([10^{-3} \quad 10^{-3} \quad 10^{-5} \quad 10^{-5}])$$

The true system is an OE representation, which is a common description of many physical processes. However, for the model set we have chosen an ARX representation which corresponds to a linear regression. The advantage of the linear regression is that it can be estimated using RLS, which is a simple and computationally efficient algorithm. However, there is a mismatch between the noise of the true system and the model, which would lead to worse estimates. To compensate this effect, data prefiltering will be considered. In the direct approach we will use RLSSVF, with an SVF (see (1.19)) of order  $n_a = 2$  and cut-off frequency  $\lambda_{\text{svf}} = 3.2$  (rad/s); note that for  $\lambda_{\text{svf}}$  we have conveniently chosen a value slightly larger than the maximum system bandwidth (see Figure 1.7). In the indirect method, for data prefiltering, we consider the same SVF discretized with the bilinear transform. The DT estimates are shown in Figure 1.8; notice that the estimates for  $\alpha_i(t_k)$  are better than the estimates for  $\beta_i(t_k)$ . At every step, the conversion from DT to CT parameters is carried out using the appropriate zero-order hold assumption. The transformed CT estimates from the indirect method are presented in Figure 1.9(a).

In the direct approach we assume a CT LTV ARX representation for the model set, *i.e.*

$$y_f^{(2)}(t_k) + a_1(t_k)y_f^{(1)}(t_k) + a_2(t_k)y_f(t_k) = b_0(t_k)u(t_k) + e(t_k) \quad (1.67)$$

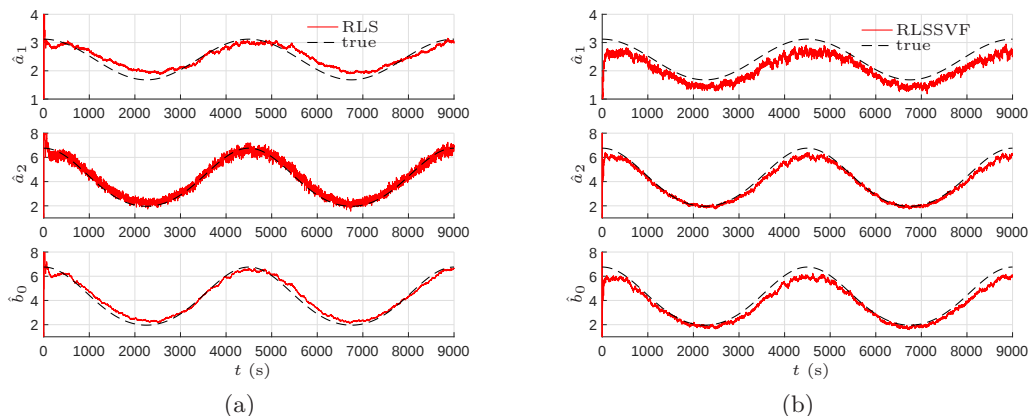


Figure 1.9: True CT parameters and CT estimates from the indirect RLS and direct RLSSVF methods.

Again, there is a mismatch between the noise of the true system and the model set. However in this case, there is no need of data prefiltering since the direct method considers a prefiltering operation in order to obtain the prefiltered time-derivatives. The model is estimated with RLSSVF-KF (1.59). We need to select the normalized covariance matrix  $Q_n$ . Since the parameters vary at a similar rate, according to Remark (1.3), we can simplify the problem and use  $Q_n = \sigma_w^2 \cdot I_3$ ; by trial and error we choose  $Q_n = 10^{-4} \cdot I_3$ . In Figure 1.9(b), the RLSSVF estimates are plotted.

Both the indirect and direct methods are able to track the CT parameters, even when the mismatch between the noise of the true system and the model introduces errors in the estimates. However, in the indirect approach we estimate first 4 DT parameters while in the direct approach 3. Therefore, in the direct approach  $Q_n$  is 1 dimension smaller, which means that it is easier to choose than in the indirect approach.

In Section 2.3.5, this example is revisited considering the right OE model and IV based methods. As we can expect, that will lead to better results.

## 1.4 Recursive identification in practice

Applying on-line identification in real-life problems is challenging, because practical and implementation issues have to be considered. In [Fridholm et al., 2016] for instance, some issues like noise effects, model errors and numerical aspects have been addressed in the context of automotive lithium-ion batteries. Literature discussing and proposing solutions to these topics are *e.g.* [Niedźwiecki, 2000] and [Åström and Wittenmark, 2008].

A very important practical issue in recursive estimation is the poor excitation situation, which has received a lot of attention in the last decades [Cao and Schwartz, 1999, 2004, Stenlund and Gustafsson, 2002, Evestedt et al., 2008]. Input richness is required for the stability and convergence of the algorithms. Under poor excitation conditions, the parameter estimates will not be accurate and in the worst case scenario they will blow up. This phenomenon is called estimator windup, and strategies to solve the problem will be presented.

Another practical issue is the choice of hyperparameters of the identification algorithms. For the Kalman filter, the hyperparameters are the covariance matrices that can be estimated using different methods like Maximum Likelihood and Expectation Maximization [Young, 2011, Bavdekar et al., 2011]. In this thesis, the former is adapted to the direct CT identification scenario.

Regarding implementation issues, we will address the numerical aspects of the estimation algorithms. In batch estimation, standard tools to improve the conditioning of the least squares problem are the singular value decomposition and QR-factorization [Ljung, 1999]. In recursive estimation, which is typically implemented in microcontrollers, this issue may be more critical due to roundoff errors introduced as a consequence of poor machine precision. Several solutions, like square root filtering (see *e.g.* [Grewal and Andrews, 2015]), have been proposed to deal with this problem. The use of such methods will be discussed in the particular context of CT identification. Additionally other implementation issues will be tackled.

## 1.5 Applications in recursive identification

Applications in which recursive schemes are used can be divided in the following categories according to [Gustafsson, 2000, p. 4]:

- Surveillance and parameter tracking. For example, monitoring and adaptive control are part of this category.
- State estimation. Observer-based controller, like Linear Quadratic Gaussian (LQG) controllers belongs to this category.
- Fault diagnosis. The use of detection together with isolation techniques, usually called diagnosis, is part of this area.

A precise definition of some of the terms mentioned above, like surveillance and monitoring, that are related to fault detection, can be found in [Gustafsson, 2000, p. 475]. It is important to keep in mind, that the evaluation criteria of the methods depends on the application [Gustafsson, 2000]. Direct CT identification methods proposed in this thesis are suitable for monitoring physical parameters. The goal in such an application is to have small tracking errors, while in other it might be more important to have small prediction errors.

There are many successful applications in which recursive estimation methods have been used. In control systems for instance, some applications using an ABB adaptive controller are presented in [Åström and Wittenmark, 2008]. In adaptive regulation, where the aim is to attenuate time-varying disturbances, some applications are given in [Landau et al., 2011]. Regarding fault detection/diagnosis, examples of applications are presented in [Isermann, 1993, Sbarbaro et al., 2008].

The application that has motivated our research is an electronic throttle control (ETC), which is a system used in automobiles to regulate air flow in the internal combustion engine. These devices have been intensively studied in the last two decades (see [Pavković et al., 2006, di Bernardo et al., 2010, Zhang et al., 2015] and references therein). In simple terms, the ETC works as follows: the airflow is regulated by a throttle valve which is actuated by a DC servomotor that is regulated by a controller. This system is time-varying because of aging effects and temperature variations. In Chapter 4, our aim is to use the developed techniques to recursively estimate the physical parameters of a throttle valve model, that could be useful for monitoring, an adaptive controller or a fault diagnosis mechanism. Since we are interested in the physical parameters of the system, we focus on direct CT model identification.

## 1.6 Organization of this thesis and contributions

In CT LTI system identification, IV based approaches are an efficient tool to reduce bias in the estimates. In particular, the SRIVC method can deliver optimal (unbiased

and minimum variance) estimates. In CT LTV system identification, IV allows us to obtain more accurate models. In Chapter 2, the goal is to extend IV based methods to the estimation of LTV systems operating both in open loop and closed loop. Through numerical examples we show the overall good performance of these algorithms.

In practice, recursive identification of DT time-varying systems is challenging, because several practical aspects and implementation issues have to be considered. In the recursive algorithms for CT LTV systems that we proposed in Chapter 2, similar practical aspects and implementation issues appear. The goal of Chapter 3 is to present suitable solutions to some of them. Regarding practical aspects, we address the problem of estimating hyperparameters of the presented algorithms. Another important aspect is the problem of poor excitation. Implementation issues which lead to improvement in numerical aspects are also discussed.

Chapter 4 is dedicated to two applications. The first one is a real-life electronic circuit which has been recently proposed as a benchmark for both LPV and LTV model identification [Lataire et al., 2015]. The second one is the throttle valve where it is assumed that the model parameters vary due to the temperature. Since there are nonlinearities in the system, the identification method proposed in Chapter 2 has to be adapted to this particular case.

## 1.7 Publications

A. Padilla, H. Garnier, and M. Gilson. Version 7.0 of the CONTSID toolbox, *17th IFAC Symposium on System Identification*, Beijing China, October 2015.

A. Padilla, H. Garnier, P. C. Young, J. I. Yuz, Real-time identification of continuous-time slowly linear time-varying systems, *55th IEEE Conference on Decision and Control*, Las Vegas US, December 2016.

A. Padilla, H. Garnier, P. C. Young, J. I. Yuz, Recursive online IV method for identification of continuous-time slowly time-varying models in closed loop, *IFAC World Congress*, Toulouse France, July 2017.

## Chapter 2

# Instrumental variable based methods

### 2.1 Introduction

Instrumental variable (IV) methods, studied in statistics, econometrics and engineering, have a long history. One of the first publications in engineering is [Wong and Polak, 1967]. The literature on this technique is extensive (see *e.g* the books [Söderström and Stoica, 1983, Young, 1984, Ljung, 1999] and the references therein). Recent studies are for instance [Gilson and Van den Hof, 2005, Laurain et al., 2011, Tóth et al., 2012]. The popularity of IV methods lies in the simplicity of some of its variants together with superior statistical properties in comparison with the least squares.

In CT LTI system identification, it is well-known that LSSVF, although simple, is unsatisfactory because the resultant parameter estimates are asymptotically biased. Instrumental variable methods are suitable approaches to deal with bias reduction. Two approaches, mentioned briefly in Chapter 1, are:

- Instrumental variable state-variable filter (IVSVF).
- Simplified refined instrumental variable method of estimation for continuous-time models (SRIVC).

In CT LTV systems, IV based methods allow us to obtain more accurate models. In this chapter, the methods IVSVF and SRIVC are extended to LTV systems, both for open loop and closed loop identification.

The contents of this chapter consists basically in two parts; the first one dedicated to open-loop identification and the second to closed-loop identification. The former is partly based on [Padilla et al., 2016], and it is organized as follows: In Section 2.2, we address the identification of linear time-invariant (LTI) systems. First we recall the optimal IV solution and then the IVSVF and SRIVC methods. Afterwards we present the recursive algorithms for the LTI case. Section 2.3 is dedicated to the identification of linear-time varying models. After formulating the problem, we discuss how to handle the problem of the time-derivatives. Then, the recursive IV based methods for the LTV case are developed. In Section 2.3.4 we discuss measures to assess the estimator performance. We finish the first part with two numerical examples.

The second part, dedicated to the closed-loop case, is based on [Padilla et al., 2017] and it is organized as follows: In Section 2.4.1, the identification problem is formulated. Then, in Section 2.4.2, the optimal closed loop IV solution is revisited. In Section 2.4.3, the closed-loop IV identification methods for LTI models are reviewed. Afterwards, the recursive algorithms for the LTI case are developed. The proposed closed-loop IV based

identification methods for linear CT slowly time-varying systems are given in Section 2.4.5 and afterwards a numerical example is presented in Section 2.4.6.

## 2.2 Open-loop identification of LTI models

### 2.2.1 Optimal IV estimation

Our goal is to find a model for the CT LTI OE system (1.1). In Chapter 1, we propose to represent  $\mathcal{S}$  with a CT LTI ARX model and estimate it using LSSVF. However, the LSSVF estimate is biased because  $\varphi_f(t_k)$  and  $v_f(t_k)$  are correlated [Garnier, 2015], since both terms are correlated with  $e(t_k)$ .

In this thesis, we propose to solve this problem with IV techniques; arguments to favor this choice have been given in Section 1.2.2. Let us recall the optimal IV method where the estimate is defined by

$$\hat{\theta} = \arg \min_{\theta \in \mathbb{R}^{n_\theta}} \frac{1}{N} \left\| \left[ \sum_{k=1}^N F(p)\zeta(t_k)F(p)\varphi^T(t_k) \right] \theta - \left[ \sum_{k=1}^N F(p)\zeta(t_k)F(p)y^{(n_a)}(t_k) \right] \right\|_W^2 \quad (2.1)$$

where  $F(p)$  is a stable prefilter and  $\|x\|_W^2 = x^T W x$ , with  $W$  a positive definite weighting matrix.  $\zeta(t_k)$  is the instrument vector that can be computed in different ways. If  $\mathcal{S} \in \mathcal{M}$ , the estimates (2.1) are consistent under the following conditions [Söderström and Stoica, 1983]<sup>1</sup>:

C1.  $\bar{\mathbb{E}}\{F(p)\zeta(t_k)F(p)\varphi^T(t_k)\}$  is full column rank.

C2.  $\bar{\mathbb{E}}\{F(p)\zeta(t_k)F(p)v_o(t_k)\} = 0$ .

with  $v_o(t_k) = A_o(p)e_o(t_k)$ . Optimal estimates, *i.e.* unbiased and minimum variance estimates, are obtained if the following additional conditions are satisfied (see [Tóth et al., 2012] and references therein)

C3.  $W = I$

C4.  $F(p) = \frac{1}{A_o(p)}$

C5. The instrument vector is computed using the auxiliary model

$$\zeta(t_k) = \begin{bmatrix} -x^{(n_a-1)}(t_k) & \dots & -x(t_k) & u^{(n_b)}(t_k) & \dots & u(t_k) \end{bmatrix}^T \quad (2.2)$$

where

$$x(t) = \frac{B_o(p)}{A_o(p)} u(t) \quad (2.3)$$

It is important to notice that the optimal solution requires knowledge of the true system, *i.e.* the optimal instrument (2.2) and optimal auxiliary model (2.3) are built assuming that the true system is known.

---

<sup>1</sup>The notation  $\bar{\mathbb{E}}[\cdot] = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{k=1}^N \mathbb{E}[\cdot]$  is adopted from the prediction error framework of [Ljung, 1999].

### 2.2.2 Off-line IV algorithms

A basic suboptimal solution to (2.1) is the IVSVF method. This is a two-step approach with  $F(p)$  defined in (1.19). In the first step, LSSVF estimates  $\hat{\theta}_1$  are computed from (1.32). In the second step, the estimates are obtained using the instrumental variable estimator, *i.e.*

$$\hat{\theta}_2 = \left[ \sum_{k=1}^N \zeta_f(t_k, \hat{\theta}_1) \varphi_f^T(t_k) \right]^{-1} \cdot \left[ \sum_{k=1}^N \zeta_f(t_k, \hat{\theta}_1) y_f^{(n_a)}(t_k) \right] \quad (2.4)$$

where  $y_f^{(n_a)}(t_k)$  and  $\varphi_f(t_k)$  are given in (1.25). The filtered instrument  $\zeta_f(t_k, \hat{\theta}_1)$  is defined through the instrument

$$\zeta(t_k, \hat{\theta}_1) = \begin{bmatrix} -\hat{x}^{(n_a-1)}(t_k) & \dots & -\hat{x}(t_k) & u^{(n_b)}(t_k) & \dots & u(t_k) \end{bmatrix}^T \quad (2.5)$$

with  $\hat{x}(t_k)$  given by the auxiliary model

$$\hat{x}(t_k) = \frac{B(p, \hat{\theta}_1)}{A(p, \hat{\theta}_1)} u(t_k) \quad (2.6)$$

Then,

$$\begin{aligned} \zeta_f(t_k, \hat{\theta}_1) &= F(p) \zeta(t_k, \hat{\theta}_1) \\ &= \begin{bmatrix} -\hat{x}_f^{(n_a-1)}(t_k) & \dots & -\hat{x}_f(t_k) & u_f^{(n_b)}(t_k) & \dots & u_f(t_k) \end{bmatrix}^T \end{aligned} \quad (2.7)$$

An optimal IV method for CT OE models is the SRIVC method. This is an iterative method, where both the prefilter and the instruments are updated in each iteration based on the parameter estimates obtained at the previous iteration. The SRIVC estimate at the  $i$ th iteration is given by

$$\hat{\theta}^i = \left[ \sum_{k=1}^N \zeta_f(t_k, \hat{\theta}^{i-1}) \varphi_f^T(t_k, \hat{\theta}^{i-1}) \right]^{-1} \cdot \left[ \sum_{k=1}^N \zeta_f(t_k, \hat{\theta}^{i-1}) y_f^{(n_a)}(t_k, \hat{\theta}^{i-1}) \right] \quad (2.8)$$

In this case, the filtered linear regression (1.24) is obtained using the following adaptive prefilter

$$F(p, \hat{\theta}^{i-1}) = \frac{1}{\hat{A}(p, \hat{\theta}^{i-1})} \quad (2.9)$$

Therefore,

$$y_f^{(n_a)}(t_k, \hat{\theta}^{i-1}) = p^{n_a} F(p, \hat{\theta}^{i-1}) y(t_k) \quad (2.10a)$$

$$\varphi_f^T(t_k, \hat{\theta}^{i-1}) = F(p, \hat{\theta}^{i-1}) \varphi^T(t_k) \quad (2.10b)$$

The filtered instrument  $\zeta_f(t_k, \hat{\theta}^{i-1})$  is defined through the instrument

$$\zeta(t_k, \hat{\theta}^{i-1}) = \begin{bmatrix} -\hat{x}^{(n_a-1)}(t_k) & \dots & -\hat{x}(t_k) & u^{(n_b)}(t_k) & \dots & u(t_k) \end{bmatrix}^T \quad (2.11)$$

with  $\hat{x}(t_k)$  defined by the auxiliary model

$$\hat{x}(t_k) = \frac{\hat{B}(p, \hat{\theta}^{i-1})}{\hat{A}(p, \hat{\theta}^{i-1})} u(t_k) \quad (2.12)$$

| Method          | LS                                   | IV                                                                                    |                                                                                               |
|-----------------|--------------------------------------|---------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------|
|                 | LSSVF                                | IVSVF                                                                                 | SRIVC                                                                                         |
| Prefilter       | $F(p) = \frac{1}{p + \lambda_{svf}}$ |                                                                                       | $F(p) = \frac{1}{\hat{A}(p, \hat{\theta}^{i-1})}$                                             |
| Auxiliary model | none                                 | $\hat{x}(t_k) = \frac{\hat{B}(p, \hat{\theta}_1)}{\hat{A}(p, \hat{\theta}_1)} u(t_k)$ | $\hat{x}(t_k) = \frac{\hat{B}(p, \hat{\theta}^{i-1})}{\hat{A}(p, \hat{\theta}^{i-1})} u(t_k)$ |

Figure 2.1: Summary of linear filter methods.

Then,

$$\begin{aligned} \zeta_f(t_k, \hat{\theta}^{i-1}) &= F(p, \hat{\theta}^{i-1}) \zeta(t_k, \hat{\theta}^{i-1}) \\ &= \begin{bmatrix} -\hat{x}_f^{(n_a-1)}(t_k) & \dots & -\hat{x}_f(t_k) & u_f^{(n_b)}(t_k) & \dots & u_f(t_k) \end{bmatrix}^T \end{aligned} \quad (2.13)$$

The three linear filter methods presented so far are summarized in Figure 2.1. We can see for each approach the corresponding prefilter and auxiliary model.

**Remark 2.1** *In a more general case, the measurement noise in  $\mathcal{S}$  might be colored. A reliable solution is then to use the RIVC method [Young, 2011]. Nevertheless, the SRIVC algorithm has the advantage of its much simpler implementation in this recursive estimation context and it performs optimally if the additive noise is white, i.e. estimates are consistent and asymptotically efficient. Moreover, if the noise is colored, with rational spectral density, then the estimates remain consistent, thanks to the inherent instrumental variable mechanism; and although they do not have minimum variance, experience has shown that the estimates are often relatively efficient (i.e. they have low but not minimum variance [Young, 2011, Garnier, 2015]). Taking into account the trade-off between accuracy and implementation issues, the recursive version of both IVSVF and SRIVC are considered in this work.*

### 2.2.3 Recursive instrumental variable methods

The recursive version of IVSVF (RIVSVF) is the following algorithm, which is based on the methodology used in the DT case, *i.e.* in this case considering the recursive instrumental variable method (see *e.g.* [Young, 2011]):

$$\hat{\theta}(t_k) = \hat{\theta}(t_{k-1}) + L(t_k) \varepsilon(t_k) \quad (2.14a)$$

$$\varepsilon(t_k) = y_f^{(n_a)}(t_k) - \varphi_f^T(t_k) \hat{\theta}(t_{k-1}) \quad (2.14b)$$

$$L(t_k) = \frac{P(t_{k-1}) \zeta_f(t_k)}{1 + \varphi_f^T(t_k) P(t_{k-1}) \zeta_f(t_k)} \quad (2.14c)$$

$$P(t_k) = P(t_{k-1}) - L(t_k) \varphi_f^T(t_k) P(t_{k-1}) \quad (2.14d)$$

where  $y_f^{(n_a)}(t_k)$  and  $\varphi_f(t_k)$  are defined in (1.25). However, note that the CT filtering operations involved in (1.25) in the off-line case are different than in the recursive case. This implementation issue is discussed later in Section 3.1.3. The filtered instrument  $\zeta_f(t_k)$  is defined through the instrument

$$\zeta(t_k, \hat{\theta}(t_{k-1})) = \begin{bmatrix} -\hat{x}^{(n_a-1)}(t_k) & \dots & -\hat{x}(t_k) & u^{(n_b)}(t_k) & \dots & u(t_k) \end{bmatrix}^T \quad (2.15)$$

with  $\hat{x}(t_k)$  given by the auxiliary model

$$\hat{A}(p, \hat{\theta}(t_{k-1}))\hat{x}(t_k) = \hat{B}(p, \hat{\theta}(t_{k-1}))u(t_k) \quad (2.16)$$

Then,

$$\begin{aligned} \zeta_f(t_k) &= F(p)\zeta(t_k, \hat{\theta}(t_{k-1})) \\ &= \begin{bmatrix} -\hat{x}_f^{(n_a-1)}(t_k) & \dots & -\hat{x}_f(t_k) & u_f^{(n_b)}(t_k) & \dots & u_f(t_k) \end{bmatrix}^T \end{aligned} \quad (2.17)$$

For the sake of simplicity, hereafter and whenever necessary, the parameter dependency will be sometimes omitted, as it is the case for  $\zeta_f(t_k)$  in (2.17), which should be  $\zeta_f(t_k, \hat{\theta}(t_{k-1}))$ . Contrary to the off-line case, where the dependency is w.r.t. an estimate of a previous step, here it is w.r.t. an estimate of a previous recursion.

For the recursive version of SRIVC (RSRIVC), the algorithm is given by (2.14), but with

$$y_f^{(n_a)}(t_k) = p^{n_a} F(p, \hat{\theta}(t_{k-1}))y(t_k) \quad (2.18a)$$

$$\varphi_f^T(t_k) = F(p, \hat{\theta}(t_{k-1}))\varphi^T(t_k) \quad (2.18b)$$

where the adaptive prefilter is

$$F(p, \hat{\theta}(t_{k-1})) = \frac{1}{\hat{A}(p, \hat{\theta}(t_{k-1}))} \quad (2.19)$$

From  $\zeta(t_k)$  given in (2.15), we define the filtered instrument  $\zeta_f(t_k)$  by

$$\begin{aligned} \zeta_f(t_k) &= F(p, \hat{\theta}(t_{k-1}))\zeta(t_k, \hat{\theta}(t_{k-1})) \\ &= \begin{bmatrix} -\hat{x}_f^{(n_a-1)}(t_k) & \dots & -\hat{x}_f(t_k) & u_f^{(n_b)}(t_k) & \dots & u_f(t_k) \end{bmatrix}^T \end{aligned} \quad (2.20)$$

It should be noted that in this case, not only  $\hat{x}(t_k)$ , but also the prefilter (2.19) depend on the estimate  $\hat{\theta}(t_{k-1})$ , meaning actually that all prefiltered variables depend on  $\hat{\theta}(t_{k-1})$ . Notice also that in off-line identification with SRIVC, the auxiliary model and prefilter depend on an estimate of a previous iteration, whereas in the recursive case they depend on an estimate of a previous recursion.

Both RIVSVF and RSRIVC are initialized with RLSSVF. The transition from the latter to the IV based methods can be manually set by the user at a time instant that we denote  $t_{s,iv}$ . Techniques that allow us to perform this transition automatically will be discussed in Chapter 3.

## 2.3 Open-loop identification of LTV models

In this section, the aim is to extend these IV approaches presented previously to real-time estimation of linear time-varying systems.

### 2.3.1 Problem formulation

Let us consider a CT LTV OE system with input  $u$  and output  $y$ . The data generating system is given by

$$\mathcal{S} \begin{cases} A_o(p, t)x(t) = B_o(p, t)u(t) \\ y(t_k) = x(t_k) + e_o(t_k) \end{cases} \quad (1.49)$$

where  $e_o(t_k)$  is a zero-mean DT white noise sequence. We assume that the system (1.49) can be represented by the model set

$$\mathcal{M} \begin{cases} A(p, t, \theta)x(t) = B(p, t, \theta)u(t) \\ y(t_k) = x(t_k) + e(t_k) \end{cases} \quad (2.21)$$

$A(p, t, \theta)$  and  $B(p, t, \theta)$  are the following polynomials with time variable parameters:

$$\begin{aligned} B(p, t, \theta) &= b_0(t)p^{n_b} + b_1(t)p^{n_b-1} + \dots + b_{n_b}(t) \\ A(p, t, \theta) &= p^{n_a} + a_1(t)p^{n_a-1} + \dots + a_{n_a}(t) \end{aligned}$$

where  $n_a \geq n_b$  and  $e(t_k)$  is a zero-mean DT white noise. The time-varying parameters can be gathered in the parameter vector  $\theta(t)$ , *i.e.*

$$\theta(t) = \begin{bmatrix} a_1(t) & \dots & a_{n_a}(t) & b_0(t) & \dots & b_{n_b}(t) \end{bmatrix}^T \quad (2.22)$$

The following assumptions are supposed to be satisfied:

A1. The system and the model set have CT LTV OE representation, with polynomial degrees  $n_a$  and  $n_b$  *a priori* known.

A2. The true parameter vector  $\theta_o(t)$  is slowly time-varying in the sense that in a local window with time interval  $s \in [t_i, t_f]$ , it can be locally approximated by a constant parameter vector  $\theta_c$ , *i.e.*

$$\|\dot{\theta}_o(t)\| \leq \epsilon_\theta \quad \text{such that} \quad \|\theta_o(s) - \theta_c\| \leq \epsilon_{\Delta\theta} \quad (2.23)$$

where  $\epsilon_\theta$  and  $\epsilon_{\Delta\theta}$  are small numbers.

A3. The input  $u$  is persistently exciting.

Then, the identification problem is to recursively estimate the time-varying parameters that characterize the model structure given by (2.21), based on sequential samples of input and output data  $Z_{N'} = \{u(t_k); y(t_k)\}_{k=1}^{N'}$ , where  $N'$  is the number of samples which increases by one with every recursion.

### 2.3.2 Handling of the time-derivative issue

As it was briefly mentioned in Section 1.3.3, multiplication with the differentiation operator  $p$  is not commutative with the time-dependent variables, *i.e.*

$$pB(p, t, \theta)u(t) \neq B(p, t, \theta)pu(t)$$

Thus, it would appear at first that we are no longer able to filter the LTV model (2.21) without introducing errors and obtain a linear regression, as in the LTI case (see (1.24)). One solution would be to model the time-varying parameters in a deterministic way such that the new parameters are constant or nearly constant, as in [Jiang and Schaufelberger, 1991]. For on-line estimation, the difficulty of such an approach lies in finding suitable deterministic models. Moreover, extra parameters are introduced and, as a consequence, larger variance of the estimates are obtained.

Alternatively, we consider Assumption (A2) in Section 2.3.1, which states that the true time-varying parameters are slowly varying in the sense that the system can be locally approximated by a constant parameter vector, *i.e.* an LTI model. Recursive estimation algorithms capable of tracking time-varying parameters have a finite memory,

meaning that the data used for identification is a sliding window in which the approximation of the LTI system is valid. Since an LTI representation does not suffer from the non-commutativity problem, then the non-commutativity aspect is negligible in the estimation of slowly LTV systems. Thus, as in the LTI case, prefilter time-derivatives can be obtained by applying a low-pass filter to the model (2.21). Analogous to the case for CT LTV ARX models (see (1.53)), applying a filter  $F(p)$  to (2.21) we obtain

$$y_f^{(n_a)}(t_k) = \varphi_f^T(t_k)\theta(t_k) + v_f(t_k) \quad (2.24)$$

where  $y_f^{(n_a)}(t_k)$  and  $\varphi_f(t_k)$  are defined in (1.25). The difference between (1.53) and (2.24) is that in the latter the filtered noise is given by

$$v_f(t_k) = F(p)A(p, t, \theta)e(t_k) \quad (2.25)$$

**Remark 2.2** *In [Dimogianopoulos and Lozano, 2001], the same reasoning is done to compute prefilter time-derivatives and estimate a CT model. Moreover, the authors provided Lemma 1 which guarantees the existence of a local LTI representation under the conditions of a bounded change ratio of the true parameters and bounded measurement noise. Lemma 1 has been derived from a exponentially weighted least squares in continuous-time in order to have an explicit dependency on the change ratio of the true parameters. However, Lemma 1 can be also expressed in DT using approximations of the derivatives and integrals involved in the formulation.*

In practice, a simple way to validate Assumption (A2) would be to compare the cost functions

$$V[\hat{\theta}(s)] = \frac{1}{N} \sum_{k=i}^f [y_f^{(n_a)}(t_k) - \varphi_f^T(t_k)\hat{\theta}(t_k)]^2 \quad (2.26)$$

with  $V[\theta_c]$ , where  $\theta_c \in \hat{\theta}(s)$ . Then, Assumption (A2) is valid, if

$$\frac{V[\theta_c]}{V[\hat{\theta}(s)]} \approx 1 \quad (2.27)$$

**Remark 2.3** *In tracking analysis it can be distinguished between transient mode and tracking mode [Rutström, 2005]. The former mode starts when the parameters jump due to either an inaccurate initialization of the recursive algorithm or an abrupt change in the system. After the transient mode, we arrive at the tracking mode which is a steady-state condition. In recursive estimation of LTI models, the estimated parameters vary abruptly in the transient mode, and then they vary slowly in the tracking mode. Thus, in the tracking mode, the non-commutativity can be neglected and an SVF can be applied to circumvent the time-derivative issue. However, in the transient mode this argument is not valid and errors in the computed estimates are introduced.*

### 2.3.3 Recursive instrumental variable methods

For the reasons exposed in Remark 2.1, let us now adapt IVSVF and SRIVC to LTV systems.

To cope with the parameter variation we assume that they can be represented by a random walk model. Then, the full model becomes:

$$\mathcal{M} \begin{cases} \theta(t_k) = \theta(t_{k-1}) + w(t_k) \\ A(p, t, \theta)x(t) = B(p, t, \theta)u(t) \\ y(t_k) = x(t_k) + e(t_k) \end{cases} \quad (2.28)$$

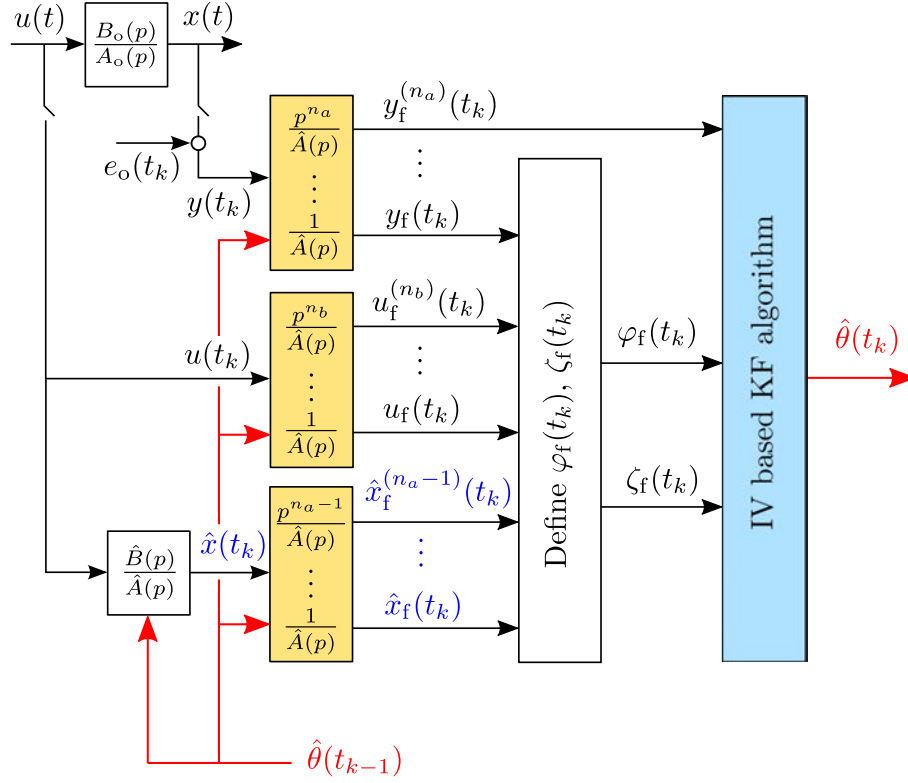


Figure 2.2: Diagram of Kalman filter based RSRIVC.

The parameters of model (2.28) can be estimated based on the methodology used in the DT case, *i.e.* in this case considering the instrumental variable based Kalman filter (see [Young, 2011, p. 298]). The algorithm is given by

*Prediction step:*

$$\hat{\theta}(t_k|t_{k-1}) = \hat{\theta}(t_{k-1}) \quad (2.29a)$$

$$P(t_k|t_{k-1}) = P(t_{k-1}) + Q_n \quad (2.29b)$$

*Correction step:*

$$\hat{\theta}(t_k) = \hat{\theta}(t_k|t_{k-1}) + L(t_k)\varepsilon(t_k) \quad (2.29c)$$

$$\varepsilon(t_k) = y_f^{(n_a)}(t_k) - \varphi_f^T(t_k)\hat{\theta}(t_k|t_{k-1}) \quad (2.29d)$$

$$L(t_k) = \frac{P(t_k|t_{k-1})\zeta_f(t_k)}{1 + \varphi_f^T(t_k)P(t_k|t_{k-1})\zeta_f(t_k)} \quad (2.29e)$$

$$P(t_k) = P(t_k|t_{k-1}) - L(t_k)\varphi_f^T(t_k)P(t_k|t_{k-1}) \quad (2.29f)$$

where  $y_f^{(n_a)}(t_k)$ ,  $\varphi_f(t_k)$  and  $\zeta_f(t_k)$  are computed as in the recursive LTI case (see Section 2.2.3). The methods are called Kalman filter based RIVSVF and Kalman filter based RSRIVC algorithms. A diagram of the Kalman filter based RSRIVC method is presented in Figure 2.2; note that the auxiliary model and adaptive filter are shown in their operator form.

Another way to deal with the estimation of time-varying parameters is the forgetting factor approach. Based on the methodology used in the DT case, the forgetting factor

based RIVSVF/RSRIVC algorithm is defined by

$$\hat{\theta}(t_k) = \hat{\theta}(t_{k-1}) + L(t_k)\varepsilon(t_k) \quad (2.30a)$$

$$\varepsilon(t_k) = y_f^{(na)}(t_k) - \varphi_f^T(t_k)\hat{\theta}(t_{k-1}) \quad (2.30b)$$

$$L(t_k) = \frac{P(t_{k-1})\zeta_f(t_k)}{\lambda + \varphi_f^T(t_k)P(t_{k-1})\zeta_f(t_k)} \quad (2.30c)$$

$$P(t_k) = \frac{1}{\lambda} \left[ P(t_{k-1}) - L(t_k)\varphi_f^T(t_k)P(t_{k-1}) \right] \quad (2.30d)$$

It should be however reminded that a drawback of those approaches is that they are not suitable for cases where parameters vary at different rates. Moreover, as we will see in Section 3.6.1, KF based approaches are in general more robust to poor excitation. This motivates the focus on KF based methods in this work.

Both the forgetting factor and Kalman filter based routines are available in the CONTinuous-Time System IDentification (CONTSID) toolbox, which has been recently updated [Padilla et al., 2015] and can be freely downloaded <sup>2</sup>.

### 2.3.4 Measures of estimator performance

In practice, the tracking ability of the approaches can be measured only indirectly in terms of the prediction error  $\varepsilon$ . Nonetheless, in simulation the true parameters are available, which allows us to assess the performance by means of the tracking error

$$\tilde{\theta}(t_k) = \theta_o(t_k) - \hat{\theta}(t_k) \quad (2.31)$$

For  $N$  samples, the mean squared value of the parameter tracking error is given by

$$\text{MSE}[\hat{\theta}(t_k)] = \frac{1}{N} \sum_{k=1}^N \|\theta_o(t_k) - \hat{\theta}(t_k)\|^2 \quad (2.32)$$

For the  $i$ th element of  $\theta_o(t_k)$ , denoted by  $\theta_i^o(t_k)$ , we can also use the relative error defined by

$$\tilde{\theta}_{i,r}(t_k) = \left| \frac{\theta_i^o(t_k) - \hat{\theta}_i(t_k)}{\theta_i^o(t_k)} \right| \times 100 \quad (2.33)$$

By adding and subtracting  $\mathbb{E}[\hat{\theta}(t_k)]$  in (2.31), the tracking error can be split into two parts [Haykin, 2014, p. 524], the lag error  $\tilde{\theta}_l(t_k)$  and the noise error  $\tilde{\theta}_n(t_k)$ , *i.e.*

$$\tilde{\theta}(t_k) = \underbrace{\theta_o(t_k) - \mathbb{E}[\hat{\theta}(t_k)]}_{\tilde{\theta}_l(t_k)} + \underbrace{\mathbb{E}[\hat{\theta}(t_k)] - \hat{\theta}(t_k)}_{\tilde{\theta}_n(t_k)} \quad (2.34)$$

These errors cannot be minimized simultaneously due to the well-known trade-off of recursive algorithms between tracking capability and noise sensitivity [Ljung and Gunnarsson, 1990]. For tracking problems, the lag error can be compared to the standard concept of bias in LTI systems [Rutström, 2005, p. 15]. The presence of the lag error is testimony to the non-stationary nature of the environment [Haykin, 2014], meaning that in the case of time-varying parameters, the lag error is larger than in the time-invariant case. Indeed, in the latter, the lag error could be zero.

The lag error and noise error could be computed by a Monte Carlo simulation using a given input and different measurement noise realizations. Then, the expectation  $\mathbb{E}[\hat{\theta}(t_k)]$  will be replaced by the sample mean  $\bar{\hat{\theta}}(t_k)$  taken over the total number of simulations.

---

<sup>2</sup>[www.cran.univ-lorraine.fr/contsid/](http://www.cran.univ-lorraine.fr/contsid/)

Such an analysis is done in Section 2.3.5. It is then convenient to define two additional measures, the mean squared value of the lag error defined by

$$\text{MSE}_l[\hat{\theta}(t_k)] = \frac{1}{N} \sum_{k=1}^N \|\theta_o(t_k) - \bar{\hat{\theta}}(t_k)\|^2 \quad (2.35)$$

and the mean squared value of the noise error given by

$$\text{MSE}_n^i[\hat{\theta}(t_k)] = \frac{1}{N} \sum_{k=1}^N \|\bar{\hat{\theta}}(t_k) - \hat{\theta}(t_k)\|^2 \quad (2.36)$$

where the superscript  $i$  means that the computation is done w.r.t. the estimate  $\hat{\theta}(t_k)$  from the  $i$ th simulation which can be arbitrarily chosen.

**Remark 2.4** *In the identification of LTI models, IV methods are applied to reduce bias. Analogously, in the LTV case, the aim is to reduce the lag error which come at the expenses of increasing slightly the noise error. That is the well-known trade-off of recursive algorithms between tracking ability and noise sensitivity.*

### 2.3.5 Numerical example - Similar parameter variations

Example 1.3.4 is revisited here to illustrate the performance of the proposed IV based Kalman filter methods. Additionally, the lag error and noise error defined in Section 2.3.4 are analyzed.

For all the simulations, we use the same value for the hyperparameters  $Q_n$  and  $\lambda_{\text{svf}}$ . The former is a diagonal matrix, and since the parameters are varying at a similar rate we consider

$$Q_n = 10^{-4} I_3$$

where  $I_3$  is the identity matrix of dimension 3. For the SVF,  $\lambda_{\text{svf}} = 3.2$  rad/s is chosen, *i.e.* a convenient value slightly larger than the maximum bandwidth (see Figure 1.7). The IV methods, which are initialized with RLSSVF, start to operate at time instant  $t_{\text{s,iv}} = 100$  s. The value for  $t_{\text{s,iv}}$  is chosen based on the convergence of both the output error and the parameter tracking error.

We present next some results for a single experiment run out of the 100 that are considered later in a Monte Carlo simulation analysis.

#### Single experiment analysis

In the LTI case, it is known that the RLSSVF estimates are always biased due to the measurement noise. Even if the bias cannot be fully removed, it can be reduced by a proper choice of the cut off frequency  $\lambda_{\text{svf}}$ . In the LTV case, this is more difficult since the system bandwidth is varying, while the SVF bandwidth is constant. Instrumental variable methods, like RSRIVC, can be used to cope with the noise and reduce the lag error, as explained in Remark 2.4. The RLSSVF and RSRIVC estimates are compared with the true parameter in Figure 2.3, from where we can see the improvement obtained by using the IV approach. The mean squared value of the parameter tracking error is compared in Table 2.1, where we can see that the performance of RIVSVF is similar to RSRIVC, the latter being slightly better as we could expect, due to the adaptive prefiltering. Note that the value for  $\lambda_{\text{svf}}$  has been conveniently chosen; an unsuitable cut-off frequency would produce worse results.

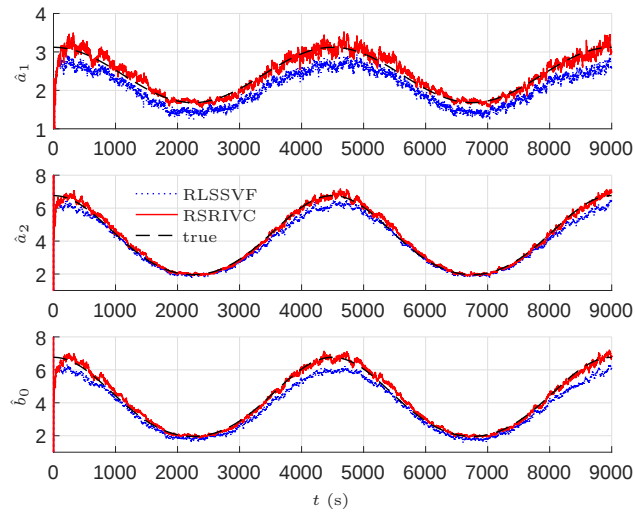


Figure 2.3: True parameters, RLSSVF estimates and RSRIVC estimates for experiment 1.

Table 2.1: Mean squared value of the parameter tracking error.

| Method | $\text{MSE}[\hat{\theta}(t_k)]$ |
|--------|---------------------------------|
| RLSSVF | 1.152                           |
| RIVSVF | 0.814                           |
| RSRIVC | 0.800                           |

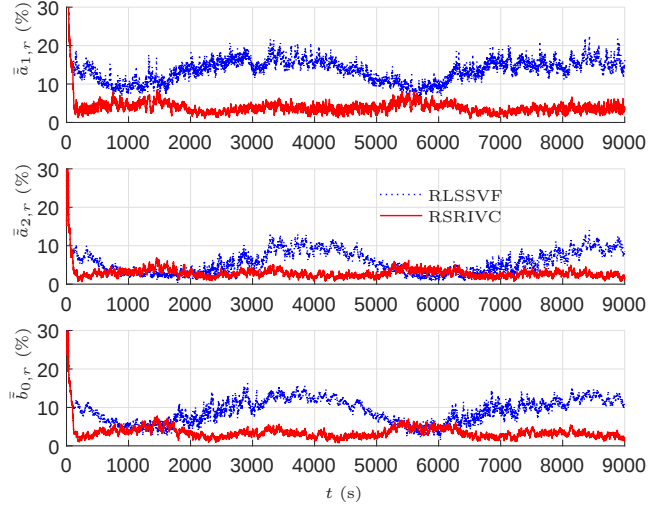


Figure 2.4: Average of the parameter relative errors for RLSSVF and RSRIVC for a Monte Carlo simulation with 100 runs.

Table 2.2: Mean squared of the lag error and mean squared of the noise error w.r.t. experiment 1.

| Method | $\text{MSE}_l[\hat{\theta}(t_k)]$ | $\text{MSE}_n^1[\hat{\theta}(t_k)]$ |
|--------|-----------------------------------|-------------------------------------|
| RLSSVF | 0.8036                            | 0.2578                              |
| RIVSVF | 0.4114                            | 0.2670                              |
| RSRIVC | 0.3998                            | 0.2672                              |

### Monte Carlo simulation analysis

The performance of these algorithms is further assessed through a Monte Carlo simulation analysis with 100 experiments. As a first performance measure, we take the relative error  $\tilde{\theta}_{i,r}$  (2.33). Averaging the relative errors over the 100 experiments, we obtain the results presented in Figure 2.4 for RLSSVF and RSRIVC. The improvement is clear from the plot.

For LTV model identification, the purpose of using IV methods is to reduce the lag error (2.34) as we can see from Figure 2.5, where its absolute value is shown for RLSSVF and RSRIVC. The absolute value of the noise error is plotted in Figure 2.6 only for the last 1000 s, to be able to distinguish the RLSSVF result from the RSRIVC result. The reduction of the lag error implies an increase of the noise error, although in terms of error magnitudes, the reduction of the former is greater than the increase in the latter. This is confirmed from Table 2.2, where the mean squared of the lag error (2.35) and mean squared of the noise error (2.36) w.r.t. experiment 1, are presented. Here we can see additionally the results for RIVSVF, whose  $\text{MSE}_l[\hat{\theta}(t_k)]$  is slightly worse than the one for RSRIVC as expected.

### 2.3.6 Numerical example - Different parameter variations

To illustrate further the performance of the proposed algorithms, let us consider a particular LTV system whose bandwidth variation in time is significant. The data generating

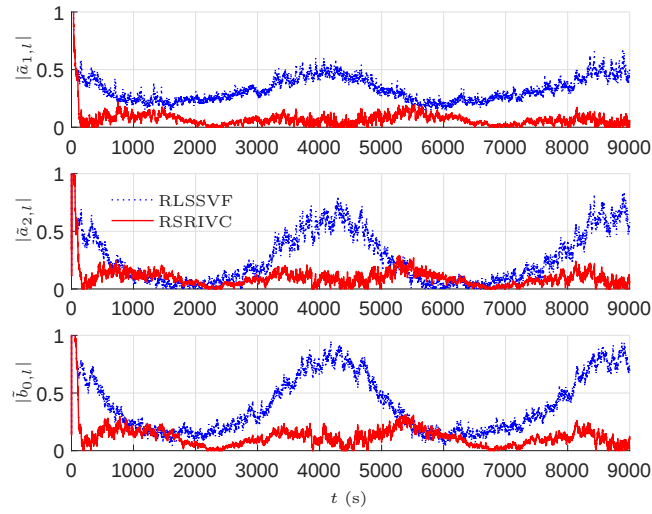


Figure 2.5: Absolute value of lag errors for RLSSVF and RSRIVC w.r.t. experiment 1.

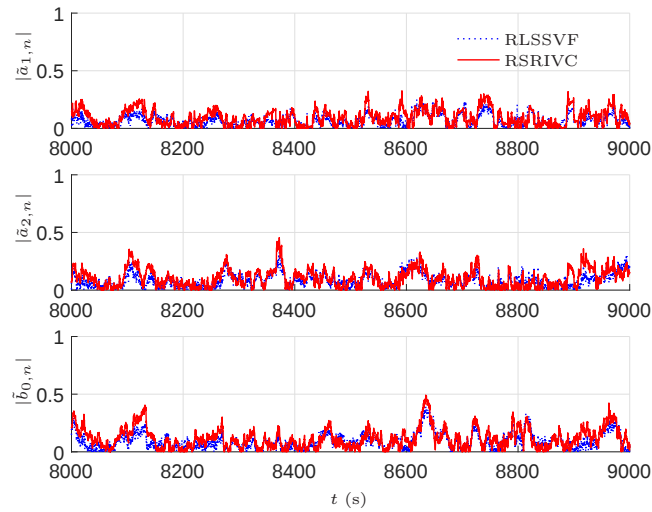


Figure 2.6: Absolute value of noise errors for RLSSVF and RSRIVC w.r.t. experiment 1.

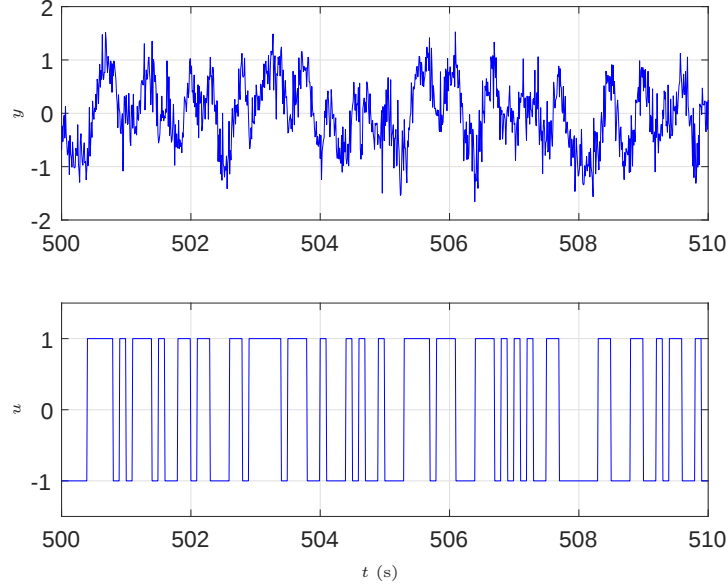


Figure 2.7: Part of the input-output data.

system is now given by the following second order CT LTV OE system.

$$\mathcal{S} \begin{cases} (p^2 + a_1^o(t)p + a_2^o(t))x(t) = b_0^o(t)u(t) \\ y(t_k) = x(t_k) + e(t_k) \end{cases} \quad (2.37)$$

where  $a_1^o(t)$  varies slowly between 5 and 45 in a linear fashion,  $b_0^o(t)$  remains constant at 200 and  $a_2^o(t)$  varies slowly as follows,

$$a_2^o(t) = 160 - 90 \cos(2\pi t/1000)$$

The sampling time is set to 0.01 s and the total simulation time is 1000 s. The input is a PRBS and the DT measurement noise is a zero-mean, Gaussian noise with constant variance 0.1. Part of the input-output data is shown in Figure 2.7.

In this example, as a consequence of the time-varying parameters, the DC gain is decreasing towards half of the simulation time; and since the noise variance is kept constant, the signal-to-noise ratio (SNR) is decreasing around half of the simulation time. For some frozen systems obtained every 100 s, the Bode diagrams are plotted in Figure 2.8, where we can see how the DC gain is varying. The step responses and bandwidths of these frozen systems are shown in Figures 2.9 and 2.10, respectively. Notice that the ratio between the maximum and minimum bandwidths is nearly 10, *i.e.* the bandwidth variation is relatively large over the total simulation time, as we mentioned.

For all the estimations in this example, the same normalized covariance matrix  $Q_n$  is used.  $Q_n$  has been obtained by a numerical search; it is set as a diagonal matrix with diagonal elements equal to  $10^{-5}$ ,  $10^{-4}$  and  $10^{-10}$  corresponding to  $a_1^o$ ,  $a_2^o$  and  $b_0^o$ , respectively. The value corresponding to  $b_0^o$  is  $10^{-10}$ , because it is assumed known that this parameter is constant. For the SVF,  $\lambda_{svf} = 16$  rad/s is chosen, *i.e.* a value slightly greater than the maximum bandwidth. The IV based methods are initialized with RLSSVF. Based on the convergence of RLSSVF, the start of the IV methods  $t_{s,iv}$  is set to 10 s. In addition, for the IV methods we consider the filter

$$M_\theta(q^{-1}) = \frac{0.04762 + 0.04762q^{-1}}{1 - 0.9048q^{-1}}$$

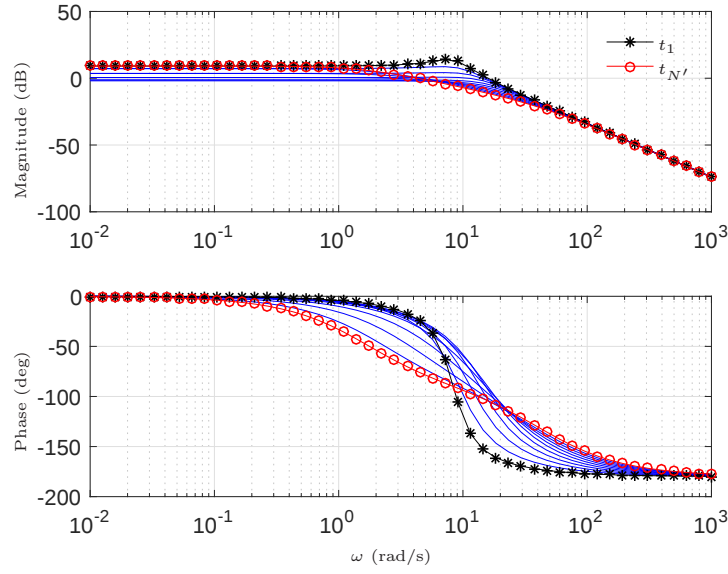


Figure 2.8: Magnitude Bode diagram for some frozen systems obtained every 100 s. The system responses for  $t = 0$  and  $t = 1000$  s are highlighted.

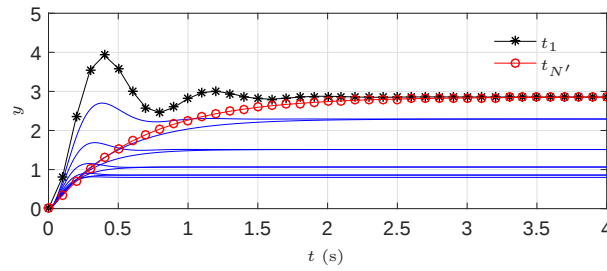


Figure 2.9: Step responses for some frozen systems obtained every 100 s. The system responses for  $t_1 = 0$  and  $t_{N'} = 1000$  s are highlighted.

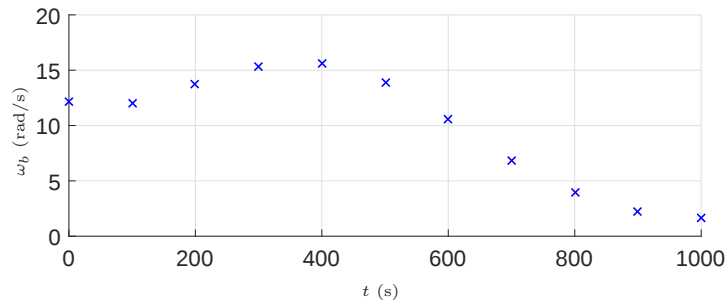


Figure 2.10: Bandwidth  $\omega_b$  for the frozen systems obtained every 100 s.

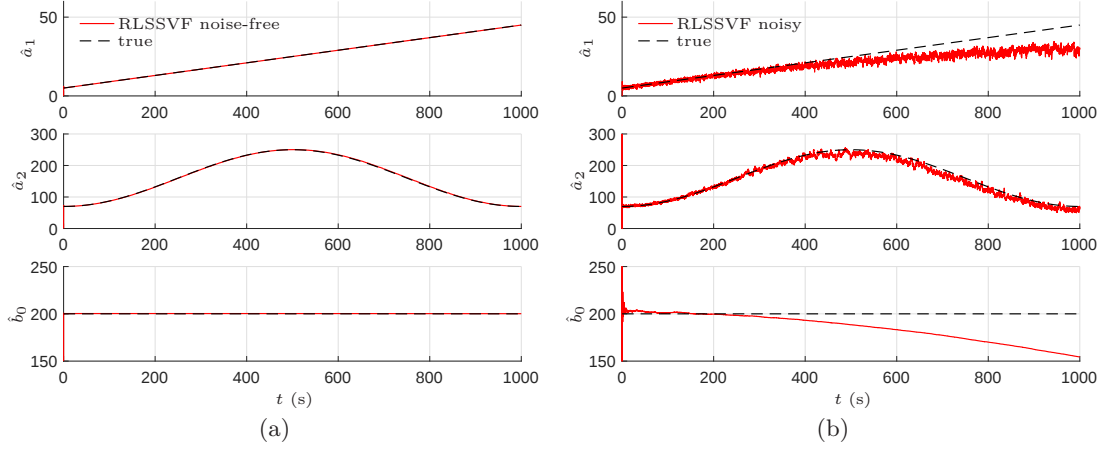


Figure 2.11: True parameters and RLSSVF estimates for the noise-free case and noisy case. (The estimates for the noise-free case are matching the true values).

This filter is used to feed back the algorithms with the estimated parameters (see more details in Section 3.2.3).

We present next some results for a single experiment run out of the 100 that are considered later in a Monte Carlo simulation analysis.

### Single experiment analysis

Firstly, the RLSSVF results are analyzed. The RLSSVF estimates are displayed in Figure 2.11 for two situations: the first a noise-free (or deterministic) case, and the second using noisy data. In the former case, since there is no noise, RLSSVF performs very well and it is difficult to distinguish between the true and estimated parameters. However, in the noisy case, RLSSVF does not provide reasonable parameter estimates, as it can be observed in Figure 2.11(b). Note that the relative errors of the three estimates are similar but this is difficult to see because the limits for the  $y$ -axes are different. Similar results have been obtained for other values of  $\lambda_{svf}$ . Notice that, in this example, the variance of the tracking error increases slightly over the central portion of the plot that can be seen in Figure 2.11. This is because the SNR is decreasing in time while the system bandwidth reduces in time (which would also require  $\lambda_{svf}$  to be lower).

To cope with the measurement noise and reduce the lag error, RIVSVF and RSRIVC are used. The improvement in performance obtained by using these IV based methods is clear from Figure 2.12. We can also see that the RSRIVC algorithm gives better results than RIVSVF at the end of the simulation time as expected, because the RSRIVC prefilter is adapted over time. Naturally, the changes of the variance of the tracking error observed in the RLSSVF case, also occur here.

### Monte Carlo simulation analysis

A more thorough analysis is done by performing a Monte Carlo simulation with 100 experiments. For RLSSVF, the average value of the relative error over the 100 experiments is shown in Figure 2.13. The same is done for the IV methods in Figure 2.14, where we can see clearly the improvements in the results. Notice also that RSRIVC is outperforming RIVSVF for the last 200 seconds. The worse results for RIVSVF are expected since in this part of the simulation, the chosen value for  $\lambda_{svf}$  (16 rad/s) is significantly larger than the system bandwidth (see Figure 2.10).

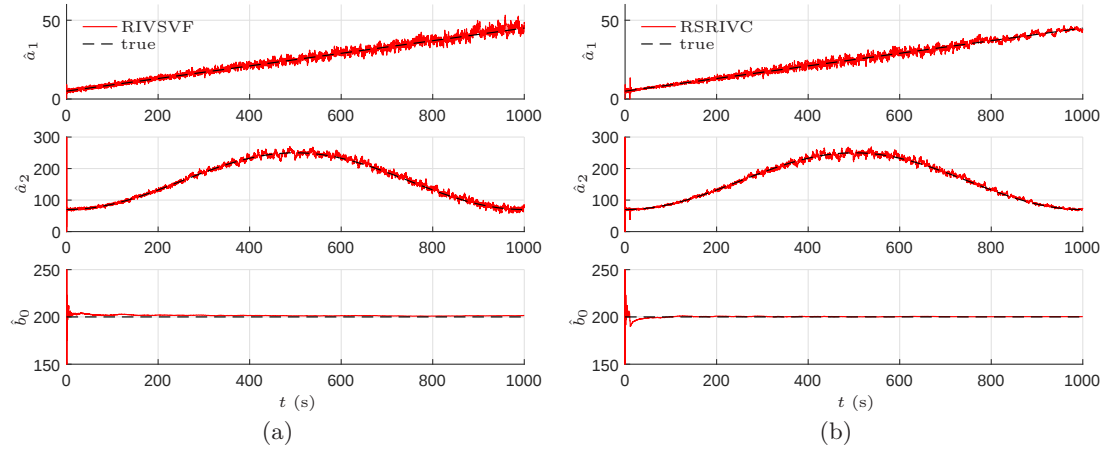


Figure 2.12: True parameters, RIVSVF estimates (a) and RSRIVC estimates (b).

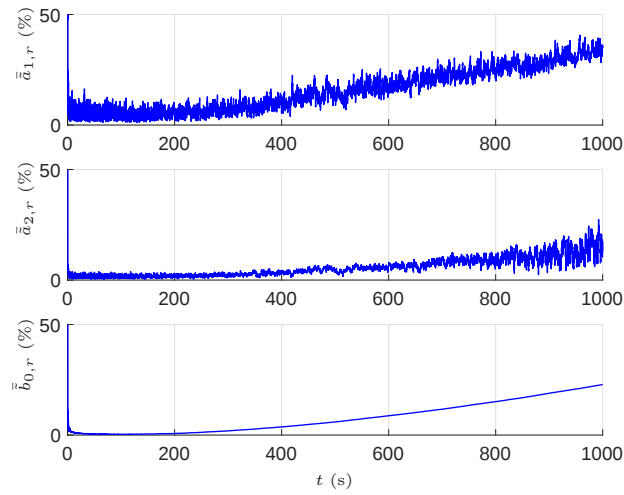


Figure 2.13: Average of the relative errors for RLSSVF for a Monte-Carlo simulation with 100 runs.

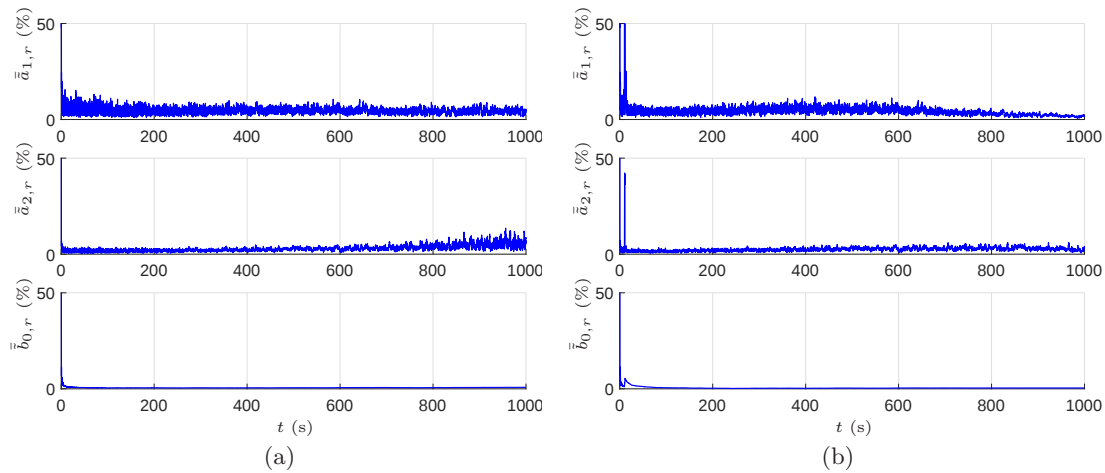


Figure 2.14: Average of the relative errors for RIVSVF (a) and RSRIVC (b) for a Monte Carlo simulation with 100 runs.

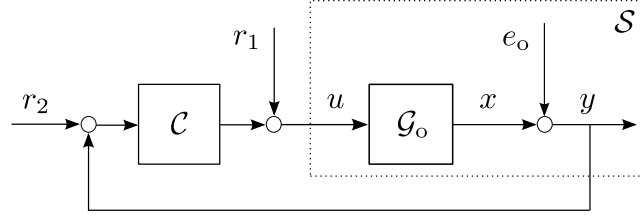


Figure 2.15: Closed-loop system.

## 2.4 Closed-loop identification

In real-life, experiments for model identification are often done in closed loop because there are economic or safety constraints, or the system is unstable. The issue with closed-loop identification is that the feedback mechanism introduces correlation between the disturbances and the input signal, yielding biased estimates if no special strategy is used. For LTI systems, instrumental variable (IV) techniques have been proposed to solve this problem both for DT models [Gilson and Van den Hof, 2005, Gilson et al., 2011] and CT models [Gilson et al., 2008]; [see also, Chapter 9 in Young, 2011]. IV have been also extended to the identification of DT linear parameter-varying models in [Tóth et al., 2012]. The recursive identification schemes developed in the previous sections are adapted here for identifying linear CT slowly time-varying systems operating in closed loop.

### 2.4.1 Problem formulation

Let us consider the closed-loop configuration from Figure 2.15 with a CT LTV OE system  $\mathcal{S}$ ,

$$\mathcal{S} \begin{cases} A_o(p, t)x(t) = B_o(p, t)u(t) \\ y(t_k) = x(t_k) + e_o(t_k) \end{cases} \quad (2.38)$$

as in the open-loop case.  $\mathcal{G}_o$  is the plant and  $e_o(t_k)$  a zero-mean DT white noise sequence. In Figure 2.15, the CT controller  $\mathcal{C}$  can be any nonlinear and/or time-varying controller. Knowing  $\mathcal{C}$ , we can compute the input  $u(t)$  as follows

$$u(t_k) = r_1(t_k) + \mathcal{C}(r_2(t_k) - y(t_k)) \quad (2.39)$$

where  $\mathcal{C}$  is the operator form of the controller and  $r_1(t_k), r_2(t_k)$  are external signals. We assume that the system can be represented by the model set  $\mathcal{M}$  given in (2.21).

Suppose that assumptions A1 and A2 defined in Section 2.3.1 for the open-loop case are satisfied. Additional assumptions are:

- A3. The controller  $\mathcal{C}$  is known.
- A4. The controller  $\mathcal{C}$  ensures BIBO stability of the closed-loop system defined by (2.38) and (2.39).
- A5. The reference signal  $r(t_k) = r_1(t_k) + \mathcal{C}r_2(t_k)$  is persistently exciting.

Then, the identification problem is to recursively estimate the time-varying parameters that characterize the model structure given by (2.21), based on the data set

$$Z_{N'} = \{r(t_k), u(t_k), y(t_k)\}_{k=1}^{N'}$$

where  $N'$  is the number of samples which increases by one with every recursion.

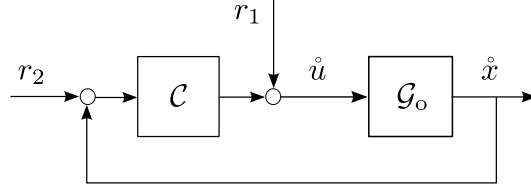


Figure 2.16: Auxiliary model.

### 2.4.2 Optimal off-line IV estimation of LTI models

In this section we assume that both the plant and the controller are LTI systems, *i.e.*

$$\mathcal{G}_o : G_o(p) = \frac{B_o(p)}{A_o(p)} \quad (2.40)$$

$$\mathcal{C} : C_c(p) = \frac{Q_c(p)}{P_c(p)} \quad (2.41)$$

with the pairs  $(A_o, B_o)$  and  $(P_c, Q_c)$  assumed to be coprime. First we address the off-line estimation problem and afterwards its recursive counterpart.

When the parameters in (2.21) are constant, they could be estimated as in the open-loop case using the LSSVF method (see (1.32)). However, the parameters will be biased because  $\varphi_f(t_k)$  and  $v_f(t_k)$  are correlated due to the filter and because  $v_f(t_k)$  is colored. Additionally, in a closed-loop configuration,  $\varphi_f(t_k)$  and  $v_f(t_k)$  are correlated due to the feedback mechanism. A solution is to use the closed-loop IV method given by (2.1) (see [Gilson et al., 2008]). The only difference w.r.t. to the open-loop case is the computation of the instruments. Then, optimal estimates, *i.e.* unbiased and minimum variance estimates, can be obtained if the conditions C1-C4 defined in Section 2.2.1 together with the following condition C5 are satisfied [Gilson et al., 2008]

C5. The instrument vector is computed using the auxiliary model from Figure 2.16 as follows

$$\zeta(t_k) = \begin{bmatrix} -\dot{x}^{(n_a-1)}(t_k) & \dots & -\dot{x}(t_k) & \dot{u}^{(n_b)}(t_k) & \dots & \dot{u}(t_k) \end{bmatrix}^T \quad (2.42)$$

where

$$A_o(p)\dot{x}(t_k) = B_o(p)\dot{u}(t_k) \quad (2.43a)$$

$$\dot{u}(t_k) = r_1(t_k) + \mathcal{C}(r_2(t_k) - \dot{x}(t_k)) \quad (2.43b)$$

### 2.4.3 Off-line IV algorithms of LTI models

Conditions C1-C3 can be readily satisfied [Söderström and Stoica, 1983]. However C4-C5 require knowledge of the true unknown system, which is the usual dilemma with optimal estimation. To circumvent this problem, different estimation algorithms have been proposed, which vary depending on the choice of the instruments  $\zeta(t_k)$ , the filter  $F(p)$  and the model structure of the true system [Gilson et al., 2008]. In the following we recall the closed-loop version of IVSVF and SRIVC presented in Section 2.2.2.

The closed-loop version of IVSVF, denoted by CLIVSVF, consists of two steps. First, LSSVF estimates  $\hat{\theta}_1$  are obtained using (1.32). In the second step, given  $N$  samples, the CLIVSVF estimates are computed from (see [Gilson et al., 2008])

$$\hat{\theta}_2 = \left[ \sum_{k=1}^N \zeta_f(t_k, \hat{\theta}_1) \varphi_f^T(t_k) \right]^{-1} \cdot \left[ \sum_{k=1}^N \zeta_f(t_k, \hat{\theta}_1) y_f^{(n_a)}(t_k) \right] \quad (2.4)$$

The difference w.r.t. the open-loop case stands in the computation of the filtered instrument  $\zeta_f(t_k, \hat{\theta}_1)$ , which is defined through the instrument

$$\zeta(t_k, \hat{\theta}_1) = \begin{bmatrix} -\hat{x}^{(n_a-1)}(t_k) & \dots & -\hat{x}(t_k) & \hat{u}^{(n_b)}(t_k) & \dots & \hat{u}(t_k) \end{bmatrix}^T \quad (2.44)$$

with  $\hat{x}(t_k)$  and  $\hat{u}(t_k)$  given by the auxiliary model

$$\hat{A}(p, \hat{\theta}_1)\hat{x}(t_k) = \hat{B}(p, \hat{\theta}_1)\hat{u}(t_k) \quad (2.45a)$$

$$\hat{u}(t_k) = r_1(t_k) + \mathcal{C}(r_2(t_k) - \hat{x}(t_k)) \quad (2.45b)$$

Then,

$$\begin{aligned} \zeta_f(t_k, \hat{\theta}_1) &= F(p)\zeta(t_k, \hat{\theta}_1) \\ &= \begin{bmatrix} -\hat{x}_f^{(n_a-1)}(t_k) & \dots & -\hat{x}_f(t_k) & \hat{u}_f^{(n_b)}(t_k) & \dots & \hat{u}_f(t_k) \end{bmatrix}^T \end{aligned} \quad (2.46)$$

In [Gilson et al., 2008], a similar algorithm, called CLIVC2, has been proposed. The difference is that in CLIVC2, a CT ARX model instead of a CT OE model, is considered.

The closed-loop version of SRIVC, denoted by CLSRIVC, has been developed in [Gilson et al., 2008]. Given  $N$  samples, the CLSRIVC estimates at the  $i$ th iteration is

$$\hat{\theta}^i = \left[ \sum_{k=1}^N \zeta_f(t_k, \hat{\theta}^{i-1}) \varphi_f^T(t_k, \hat{\theta}^{i-1}) \right]^{-1} \cdot \left[ \sum_{k=1}^N \zeta_f(t_k, \hat{\theta}^{i-1}) y_f^{(n_a)}(t_k, \hat{\theta}^{i-1}) \right] \quad (2.47)$$

As in the open-loop case,  $y_f^{(n_a)}(t_k, \hat{\theta}^{i-1})$  and  $\varphi_f(t_k, \hat{\theta}^{i-1})$  are obtained from (2.10) using the adaptive prefilter (2.9). The difference w.r.t. the open-loop case stands in the computation of the filtered instrument  $\zeta_f(t_k, \hat{\theta}^{i-1})$ , which is defined through the instrument

$$\zeta(t_k, \hat{\theta}^{i-1}) = \begin{bmatrix} -\hat{x}^{(n_a-1)}(t_k) & \dots & -\hat{x}(t_k) & \hat{u}^{(n_b)}(t_k) & \dots & \hat{u}(t_k) \end{bmatrix}^T \quad (2.48)$$

with  $\hat{x}(t_k)$  and  $\hat{u}(t_k)$  given by the auxiliary model

$$\hat{A}(p, \hat{\theta}^{i-1})\hat{x}(t_k) = \hat{B}(p, \hat{\theta}^{i-1})\hat{u}(t_k) \quad (2.49a)$$

$$\hat{u}(t_k) = r_1(t_k) + \mathcal{C}(r_2(t_k) - \hat{x}(t_k)) \quad (2.49b)$$

Then,

$$\begin{aligned} \zeta_f(t_k, \hat{\theta}^{i-1}) &= F(p, \hat{\theta}^{i-1})\zeta(t_k, \hat{\theta}^{i-1}) \\ &= \begin{bmatrix} -\hat{x}_f^{(n_a-1)}(t_k) & \dots & -\hat{x}_f(t_k) & \hat{u}_f^{(n_b)}(t_k) & \dots & \hat{u}_f(t_k) \end{bmatrix}^T \end{aligned} \quad (2.50)$$

#### 2.4.4 Recursive estimation of LTI models

The two closed-loop identification methods presented previously can be adapted for recursive estimation. As in the open-loop scenario, the algorithm is given by (2.14). The difference is only in how the filtered instrument  $\zeta_f(t_k)$  is computed.

For the recursive version of CLIVSVF, denoted by CLRIVSVF, the filtered instrument is defined through the instrument

$$\zeta(t_k, \hat{\theta}(t_{k-1})) = \begin{bmatrix} -\hat{x}^{(n_a-1)}(t_k) & \dots & -\hat{x}(t_k) & \hat{u}^{(n_b)}(t_k) & \dots & \hat{u}(t_k) \end{bmatrix}^T \quad (2.51)$$

with  $\hat{x}(t_k)$  and  $\hat{u}(t_k)$  given by the auxiliary model

$$\hat{A}(p, \hat{\theta}(t_{k-1}))\hat{x}(t_k) = \hat{B}(p, \hat{\theta}(t_{k-1}))\hat{u}(t_k) \quad (2.52a)$$

$$\hat{u}(t_k) = r_1(t_k) + \mathcal{C}(r_2(t_k) - \hat{x}(t_k)) \quad (2.52b)$$

where  $\mathcal{C}$  is the controller. Then,

$$\begin{aligned}\zeta_f(t_k) &= F(p)\zeta(t_k, \hat{\theta}(t_{k-1})) \\ &= \begin{bmatrix} -\hat{x}_f^{(n_a-1)}(t_k) & \dots & -\hat{x}_f(t_k) & \hat{u}_f^{(n_b)}(t_k) & \dots & \hat{u}_f(t_k) \end{bmatrix}^T\end{aligned}\quad (2.53)$$

For the recursive version of CLSRIVC, denoted by CLRSRIVC, the filtered instrument is given by

$$\begin{aligned}\zeta_f(t_k) &= F(p, \hat{\theta}(t_{k-1}))\zeta(t_k, \hat{\theta}(t_{k-1})) \\ &= \begin{bmatrix} -\hat{x}_f^{(n_a-1)}(t_k) & \dots & -\hat{x}_f(t_k) & \hat{u}_f^{(n_b)}(t_k) & \dots & \hat{u}_f(t_k) \end{bmatrix}^T\end{aligned}\quad (2.54)$$

with  $\zeta(t_k, \hat{\theta}(t_{k-1}))$  defined in (2.51) and the prefilter in (2.19).

### 2.4.5 Recursive estimation of LTV models

In a closed-loop configuration, tracking time-varying parameters changing at different rates can be done using the Kalman filter based RLSSVF. For the reason exposed above, these estimates would be biased. A solution is then to use the closed-loop versions of RIVSVF and RSRIVC, which are based on the Kalman filter instrumental variable algorithm given by (2.29). The difference w.r.t. the open-loop case stands in how the filtered instruments  $\zeta_f(t_k)$  are computed. For these new methods, denoted by CLRIVSVF and CLRSRIVC,  $\zeta_f(t_k)$  is defined in the previous Section 2.4.4.

### 2.4.6 Numerical example

Three recursive algorithms are evaluated: CLRIVSVF, CLRSRIVC and RLSSVF, which is used to initialize the first two approaches. The example presented in Section 2.3.6 is adapted here to the closed-loop configuration shown in Figure 2.15. Regarding the external signals (see Figure 2.15),  $r_1(t)$  is a PRBS and  $r_2(t) = 0$ .  $e_o(t_k)$  is a zero-mean DT Gaussian noise with constant variance 0.1. Remember that the ratio between the maximum and minimum bandwidths is nearly 10, *i.e.* the bandwidth variation is relatively large over the total simulation time.

CT filtering operations for the computation of prefiltered time-derivatives and instruments are implemented from their discretized counterparts. The discretized version of the PID controller is given by

$$\mathcal{C} : C_c(q^{-1}) = k_p + \frac{k_i}{1 - q^{-1}} + k_d(1 - q^{-1}) \quad (2.55)$$

with  $q^{-1}$  the backward shift operator and  $k_p = 1.79$ ,  $k_i = 13.8$ ,  $k_d = 5.83 \cdot 10^{-2}$ . For all the simulations, we use the same value for the hyperparameters  $Q_n$  and  $\lambda_{svf}$  as before.

We present next some results for a single experiment run out of the 100 that are considered later in a Monte Carlo simulation analysis.

### Single experiment analysis

In the LTI open-loop case, it is known that the RLSSVF estimates are always biased due to the noise. Even if the bias cannot be removed, it can be reduced by a proper choice of the cut off frequency  $\lambda_{svf}$ . In the open-loop LTV case, this is more difficult since the system bandwidth is varying, while the SVF bandwidth is constant, as it was illustrated in Section 2.3.6. When the system operates in closed loop there is an additional issue, namely the correlation between the input  $u(t_k)$  and the noise  $e_o(t_k)$  due to the feedback

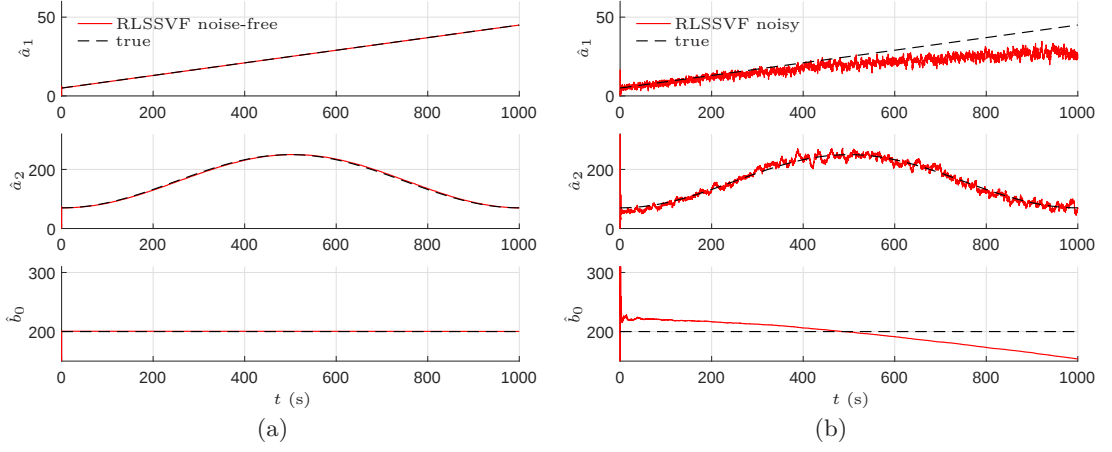


Figure 2.17: True parameters and RLSSVF estimates for the noise-free case and noisy case. (The estimates for the deterministic case are matching the true values).

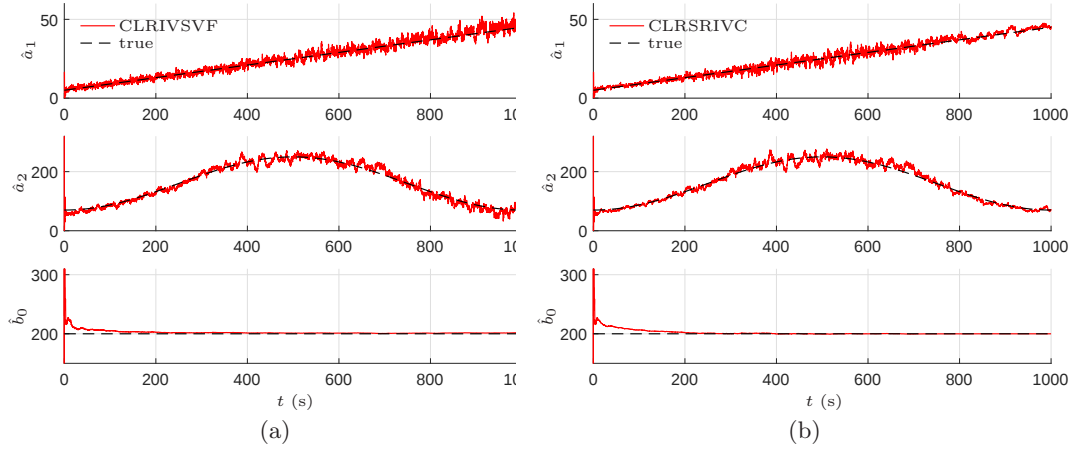


Figure 2.18: True parameters, CLRVSVF estimates and CLRSRIVC estimates.

mechanism. In the LTI case, that might result in larger bias on the estimates, while in the LTV case, it results in larger lag errors. In order to illustrate the impact of the noise on RLSSVF in the closed-loop LTV situation, we compare the noise-free case with the noisy case. From Figure 2.17, we can see that RLSSVF is able to track the parameters only in the noise-free situation. It is important to stress that the value used for  $\lambda_{svf}$  is a very good choice since it is slightly higher than the maximum system bandwidth.

To circumvent the parameter tracking problem, the two closed-loop IV approaches introduced in Section 2.4.5 have been used. From Figure 2.18 we can see that both CLRVSVF and CLRSRIVC are able to track the parameters in this situation where the parameters vary at different rates. The former approach considers a constant prefilter while in the latter the filter is automatically adapted from the previous estimates.

### Monte Carlo simulation analysis

In order to see more clearly the difference between these algorithms, a Monte Carlo simulation analysis with 100 experiments is run. For comparison purposes, the relative error (2.33) is used as a measure. Averaging the relative errors of the 100 experiments, we obtain the results presented in Figure 2.19 for RLSSVF and in Figure 2.20 for CLRVSVF and CLRSRIVC. It can be seen that towards the end of the simulation time, the best

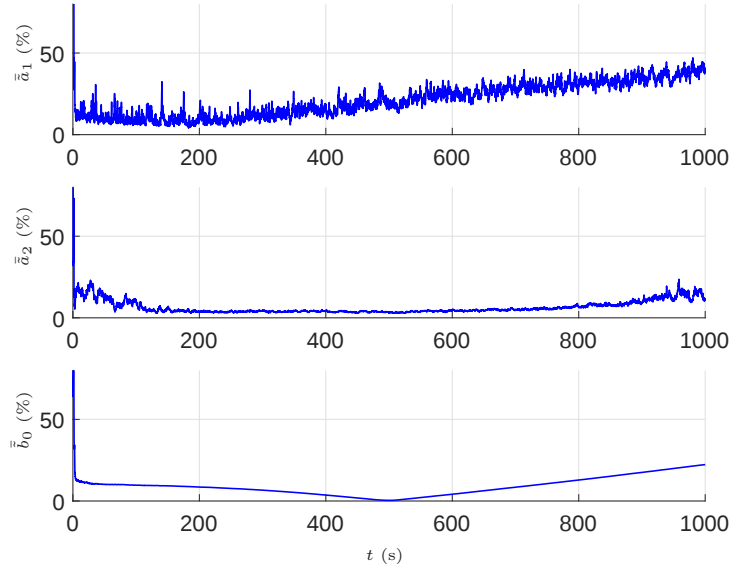


Figure 2.19: Average of the relative errors for RLSSVF for a Monte Carlo simulation with 100 runs.

results are obtained with CLRSRIVC.

## 2.5 Conclusions

In this chapter, the algorithms RIVSVF and RSRIVC have been developed for open-loop identification. Closed-loop versions have been also presented. The recursive algorithms follow from the equivalent discrete-time versions but are more complex because they require prefiltering for handling the time-derivatives. These approaches are able to track slow variations in the model parameters, based on an assumed stochastic random walk model for the parameter variations. One of the advantage of the random walk is that it is suitable when the parameters vary at different rates.

RIVSVF can provide good estimates if prior knowledge exists to define the cut-off frequency of the SVF, meaning that it can be chosen close to the true system bandwidth. However, if the system bandwidth vary too much, it will not be possible to choose a value which is suitable for every moment. With RSRIVC, this is not an issue since it considers an adaptive filter which is updated using previous estimates.

The approaches have been evaluated by means of Monte Carlo simulation analysis that demonstrates the effectiveness of the IV approaches to reduce the tracking error, in comparison with RLSSVF.

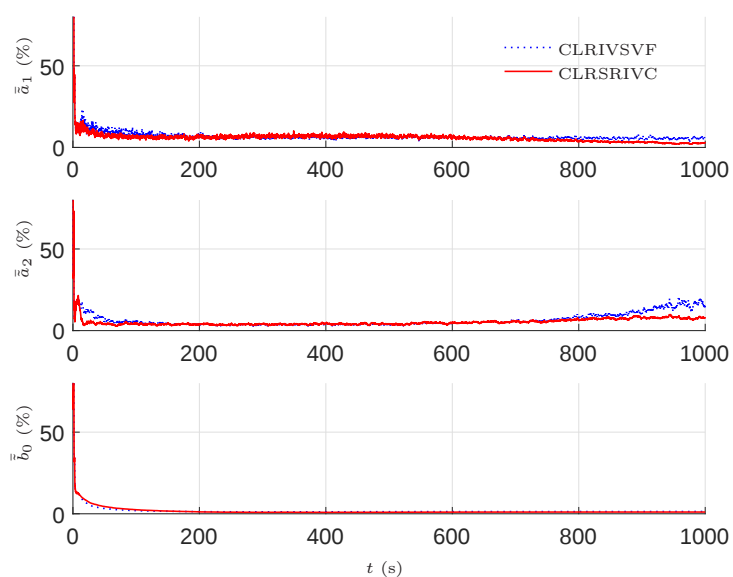


Figure 2.20: Average of the relative errors for CLRIVSVF and CLRSRIVC for a Monte Carlo simulation with 100 runs.

## Chapter 3

# Practical aspects and implementation issues

Practical aspects and implementation issues in recursive estimation are of crucial importance, and have been discussed *e.g.* in the books [Åstrom and Wittenmark, 2008] and [Niedźwiecki, 2000]. In this chapter we investigate these problems for the approaches developed in the previous chapter. The particularity of those methods is basically that they handle CT models and they use IV techniques.

The first part of this chapter is dedicated to the choice of the hyperparameters for the methods developed in Chapter 2. We will focus on RSRIVC, since the analysis for the other approaches is analogous. A diagram of RSRIVC highlighting the practical aspects and implementation issues is presented in Figure 3.1; the issues, denoted by Ii in the figure, are the following:

- I1. Choice of user hyperparameters, which are:
  - $\lambda_{svf}$ : cut-off frequency of the RLSSVF which is used to initialize RSRIVC. In Section 3.1, aspects related to the SVF, including the choice of  $\lambda_{svf}$  are discussed.
  - $t_{s,iv}$ : time instant when RSRIVC starts to operate. In Section 3.2 aspects related to IV techniques, including the choice of  $t_{s,iv}$ , are presented.
  - $Q_n$ : normalized covariance matrix of the Kalman filter. In Section 3.3, an estimation of  $Q_n$  using maximum likelihood is proposed.
- I2. Digital implementation of the CT filtering operations involved in the prefilter and auxiliary model. This is covered in Section 3.1.3.
- I3. Numerical aspects. One way to solve this implementation issue is to use square root filtering. For our methods we will see that a suitable approach is to use signal scaling. This is done in Section 3.5.
- I4. Estimator windup problem. One of the assumptions made in the theoretical developments in the previous chapter is that the input  $u(t_k)$  is persistently exciting. In practice, there might be intervals where this condition is not met. Solutions for this phenomenon, called estimator windup, will be discussed in Section 3.6.

These and some other issues will be discussed along the chapter. For instance, another practical aspect presented in Section 3.4 is how to handle the unknown initial conditions of the system that we want to identify. We will show that by properly handling these initial conditions, the overshoots that they can produce in the estimates during the initial recursions can be reduced.

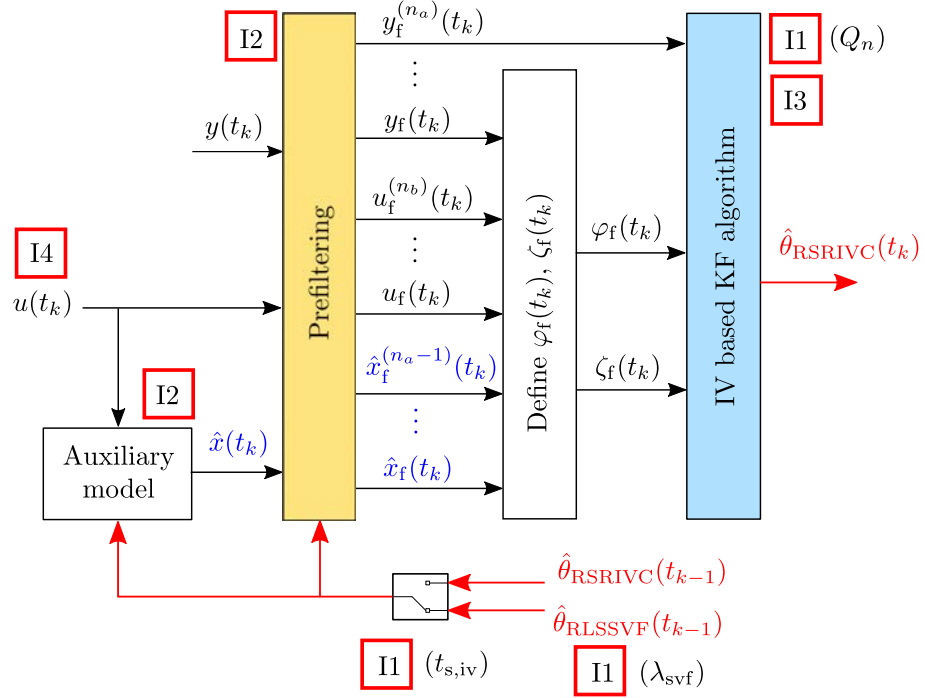


Figure 3.1: Diagram of RSRIVC with issues (Ii) regarding practical usage and implementation (see text for the description of the issues).

### 3.1 Aspects related to the SVF

#### 3.1.1 Normalized vs ordinary SVF

In the direct identification approaches presented in Chapter 2, the SVF allows us to obtain prefiltered time-derivatives. For the SVF, two forms are possible, the ordinary version (OSVF) defined in (1.19) and the normalized version (NSVF)

$$F_{\text{nsvf}}(p) = \left( \frac{\lambda_{\text{svf}}}{p + \lambda_{\text{svf}}} \right)^{n_a} \quad (3.1)$$

Thus, these forms are linked as follows

$$F_{\text{nsvf}}(p) = \lambda_{\text{svf}}^{n_a} F_{\text{osvf}}(p) \quad (3.2)$$

For the OSVF, given  $N'$  samples, the parameter covariance matrix is, up to a scaling factor, defined by

$$P_{\text{osvf}}(t_{N'}) = \sum_{k=1}^{N'} \left[ \varphi_f(t_k) \varphi_f^T(t_k) \right]^{-1} \quad (3.3)$$

with  $\varphi_f(t_k)$  defined in (1.27). Using the NSVF, the filtered regressor vector is given by

$$\begin{aligned} \varphi_{f,\text{nsvf}}^T(t_k) &= F_{\text{nsvf}}(p) \varphi_f^T(t_k) \\ &= \lambda_{\text{svf}}^{n_a} \varphi_f^T(t_k) \end{aligned} \quad (3.4)$$

Then, in this case the parameter covariance matrix is, up to a scaling factor, defined as

follows

$$\begin{aligned} P_{\text{nsvf}}(t_{N'}) &= \sum_{k=1}^{N'} \left[ \varphi_{\text{f,nsvf}}(t_k) \varphi_{\text{f,nsvf}}^T(t_k) \right]^{-1} \\ &= \frac{1}{\lambda_{\text{svf}}^{2n_a}} \sum_{k=1}^{N'} \left[ \varphi_{\text{f}}(t_k) \varphi_{\text{f}}^T(t_k) \right]^{-1} \end{aligned} \quad (3.5)$$

Thus, from (3.3) and (3.5) we obtain [Garnier et al., 1994]

$$P_{\text{osvf}}(t_{N'}) = \lambda_{\text{svf}}^{2n_a} P_{\text{nsvf}}(t_{N'}) \quad (3.6)$$

To use RLSSVF, either with OSVF or NSVF, the initial estimates  $(\hat{\theta}(t_0), P(t_0))$  have to be set. The RLSSVF estimates obtained with OSVF and  $(\hat{\theta}(t_0), P_{\text{osvf}}(t_0))$  are identical to the RLSSVF estimates obtained with NSVF and  $(\hat{\theta}(t_0), P_{\text{nsvf}}(t_0))$ .

Notice that the ordinary version could introduce a significant signal scaling, leading to numerical problems, that deteriorate the identification results<sup>1</sup>.

**Suggested choice/procedure 3.1** *In order to avoid signal scaling problems we suggest to use in general the normalized version of the SVF.*

### 3.1.2 Choice of $\lambda_{\text{svf}}$

In [Canudas de Wit, 1986], it has been shown that the parameter convergence of RLSSVF depends on the cut-off frequency of the SVF  $\lambda_{\text{svf}}$ , the sampling time and the input signal. Regarding the cut-off frequency of the SVF, the larger the value of  $\lambda_{\text{svf}}$  the quicker the convergence rate. On the other hand, the smaller the sampling time the quicker the convergence rate.

The cut-off frequency  $\lambda_{\text{svf}}$  is a user parameter that should be chosen somewhat larger than the system bandwidth. In the LTV case and especially for systems with relatively large variations of the bandwidth, the specification of  $\lambda_{\text{svf}}$  can be critical since the system bandwidth is time-varying. For its choice, the convergence aspects discussed above should be also keep in mind.

**Suggested choice/procedure 3.2** *In general, the identification of LTI should be easier than the identification of LTV models. Thus, assuming that the parameters are slowly varying, one approach to have an idea about  $\lambda_{\text{svf}}$  is to estimate local LTI models on segments of the training data. One option is to use SRIVC.*

### 3.1.3 Digital implementation of CT filtering operations

The digital implementation issues of the CT filtering operations using an SVF (see (1.20) or (1.21)) are well-known in CT model identification. They should be treated in an appropriate way since errors generated by the digital implementation can have an impact on the quality of the estimated model.

When the intersample behavior of the filter input is piecewise constant, *i.e.* a zero-order hold assumption is considered, an exact solution to the filtering operation can be obtained. This is valid both for off-line and on-line computations. When the filter input is not piecewise constant, a traditional assumption is to consider it to be linear. Then, we could use a first-order hold assumption or the bilinear (or Tustin) transform. An advantage of the latter is that it maps a stable region in the  $s$ -plane into a stable region in

---

<sup>1</sup>An analysis about signal scaling and its impact in the identification problem is presented in Section 3.5.

the  $z$ -plane [Franklin et al., 1998, p. 193]. Notice that the accuracy of these approaches depend on the sampling time, namely the faster the sampling the more accurate the filter output.

In this thesis, when the intersample behavior is assumed to be linear, the Tustin transform is considered.

**Remark 3.1** *In the off-line case, when the intersample behavior is assumed to be linear, a first-order hold assumption can be used without introducing any delay in the signal. However, in the on-line case, both the first-order hold assumption and the bilinear transform introduce a delay.*

The digital implementation of the CT filtering operations using an SVF is carried out from a time-invariant state space representation. In Appendix A, a general digital implementation considering a time-varying state space model is presented. That holds also for the LTI SVF but with constant matrices.

**Suggested choice/procedure 3.3** *When the filter input is not piecewise constant, we suggest to use the bilinear transform. In order to obtain accurate filter outputs, the sampling frequency should be as high as possible.*

## 3.2 Aspects related to the IV approaches

In this section, the discussions will be done considering only the RSRIVC method, although they will hold for RIVSVF, CLRSRIVC and CLRIVSVF as well.

### 3.2.1 Start of the IV methods

RSRIVC is initialized using RLSSVF. The switch between these algorithms takes place at some time instant  $t_{s,iv}$ , which is when RSRIVC defined in (2.29) starts to operate. The prefiltered time-derivatives and instruments are computed from the final RLSSVF estimate; afterwards, previous RSRIVC estimates are used.

The switch to (2.29) at  $t_{s,iv}$  should be done once convergence of the RLSSVF estimates is achieved. The convergence of RLSSVF could be automatically determined based on checks of variability or steady state condition of:

- $\Delta\hat{\theta}(t_k) = \hat{\theta}(t_{k-1}) - \hat{\theta}(t_k)$
- the size of the parameter covariance matrix  $P(t_k)$ , which can be measured through its trace or determinant
- the condition number of the parameter covariance matrix  $P(t_k)$
- the prediction error  $\varepsilon(t_k)$
- the output error  $\varepsilon_y(t_k)$  defined in (1.60)

For instance, the variability of  $\Delta\hat{\theta}(t_k)$  can be measured through its variance that may be computed recursively with an exponential moving average [Bittencourt et al., 2015]. The steady state condition, for instance of the output error, can be assessed using the approach proposed in [Cao and Rhinehart, 1995]. Note that the automatic detection of convergence of RLSSVF requires to set threshold values, which depend on the system.

**Suggested choice/procedure 3.4** *A simple option to set  $t_{s,iv}$  is to use RLSSVF with some training data, and to check convergence based on the prediction error or the output error. For the latter, the frequency content is lower, and therefore it might be easier to visualize convergence in a plot.*

### 3.2.2 Stability test

In off-line estimation using SRIVC, the prefilter (2.9) and the auxiliary model (2.6), depend on the estimates of the previous iteration. To ensure that both the prefilter and auxiliary model are stable, a stability test is required (see *e.g.* [Garnier, 2015]). Analogously, in on-line estimation using RSRIVC, the prefilter (2.19) and the auxiliary model (2.16) depend on the estimates of the previous recursion  $\hat{\theta}(t_{k-1})$ . Then, a stability check is also required. However, in this case it is more complicated, since instead of dealing with an LTI system, we have an LTV system. Both the prefilter and the auxiliary model can be written as an LTV state-space model

$$\mathcal{M}_{ss} \begin{cases} \dot{x}(t) = F(t)x(t) + G(t)u(t) \\ y(t) = H(t)x(t) + J(t)u(t) \end{cases} \quad (3.7)$$

Let us recall first how to check stability of (3.7). This can be done using the following lemma (see [Rugh, 1996, p. 206])

**Lemma 1** *Suppose the system (3.7) is uniform exponentially stable, and there exist finite constant  $\mu_1$ ,  $\mu_2$  and  $\mu_3$  such that for all  $t$*

$$\|G(t)\| \leq \mu_1, \quad \|H(t)\| \leq \mu_2, \quad \|J(t)\| \leq \mu_3 \quad (3.8)$$

*Then the state equation also is uniformly bounded-input, bounded-output stable.*

More details are given in Appendix B, where we recall some important theorems about LTV system stability. Assuming that  $G(t)$ ,  $H(t)$  and  $J(t)$  are bounded, *i.e.* (3.8) is fulfilled, it remains to check uniform exponential stability. That can be done in practice by solving a linear matrix inequality problem.

A more simple approach is the following. In general, uniform exponential stability cannot be characterized by the location of the eigenvalues of the matrix  $F(t)$ . However, if it is assumed that the parameters vary slowly, uniform exponential stability is guaranteed if the eigenvalues of  $F(t)$  are in the left half plane [Ilchmann et al., 1987]. Such procedure is usually considered in recursive estimation (see *e.g.* [Ljung and Söderström, 1983, p. 93]) and it is also the one adopted in this thesis.

Now we recall a stability test for  $\hat{\theta}(t_k)$  [Ljung, 1999]. Let us define first a set  $D_s$  containing stable estimates, *i.e.*

$$D_s = \{\theta \mid \mathcal{M}_{ss} \text{ has all eigenvalues in the left half plane}\} \quad (3.9)$$

If we denote the estimate before the stability check by  $\hat{\theta}^*(t_k)$ , and after it by  $\hat{\theta}(t_k)$ , then

$$\hat{\theta}(t_k) = \begin{cases} \hat{\theta}^*(t_k) & \text{if } \hat{\theta}^*(t_k) \in D_s \\ \hat{\theta}^\diamond(t_k) & \text{if } \hat{\theta}^*(t_k) \notin D_s \end{cases} \quad (3.10)$$

where  $\hat{\theta}^\diamond(t_k)$  is a stable estimate obtained

- from the previous recursion, *i.e.*  $\hat{\theta}(t_{k-1})$ , or
- by reflecting the eigenvalues of  $F(t)$  (obtained from  $\hat{\theta}^*(t_k)$ ) into the left half plane. This means that the sign of the real part of the unstable eigenvalues is changed.

### 3.2.3 Correlation between filtered instruments and filtered noise

In the batch estimation of LTI systems, consistent estimates can be obtained with SRIVC if the two conditions specified in Section 2.2.1 are fulfilled. The condition C2 is given by

$$\bar{\mathbb{E}}\{F(p, \theta_o)\zeta(t_k, \theta_o)F(p, \theta_o)v_o(t_k, \theta_o)\} = 0$$

or

$$\bar{\mathbb{E}}\{\zeta_f(t_k, \theta_o)v_{o,f}(t_k, \theta_o)\} = 0$$

which means that  $\zeta_f(t_k)$  and  $v_{o,f}(t_k)$  are uncorrelated.

In the on-line case with RSRIVC, C2 takes the form

$$\begin{aligned} \bar{\mathbb{E}}\{F(p, \hat{\theta}(t_{k-1}))\zeta(t_k, \hat{\theta}(t_{k-1}))F(p, \hat{\theta}(t_{k-1}))v_o(t_k)\} = \\ \bar{\mathbb{E}}\{\zeta_f(t_k, \hat{\theta}(t_{k-1}))v_{o,f}(t_k, \hat{\theta}(t_{k-1}))\} = 0 \end{aligned} \quad (3.11)$$

Then, (3.11) holds asymptotically as  $\hat{\theta}(t_k)$  gets close to the true value  $\theta_o(t_k)$ . When  $\hat{\theta}(t_k)$  is not close to  $\theta_o(t_k)$ ,  $v_{o,f}(t_k, \hat{\theta}(t_{k-1}))$  is colored, meaning that it will be correlated with its previous value  $v_{o,f}(t_{k-1}, \hat{\theta}(t_{k-2}))$ . On the other hand, the filtered instrument  $\zeta_f(t_k, \hat{\theta}(t_{k-1}))$  depends on the previous estimates  $\hat{\theta}(t_{k-1})$ , which at the same time depend on  $v_{o,f}(t_{k-1}, \hat{\theta}(t_{k-2}))$  through the filtered output  $y_f^{(n_a)}(t_k)$ . Thus, (3.11) will be slightly violated. This is analogous to what happens in DT identification when IV techniques are used (see [Ljung and Söderström, 1983, p. 237] and also [Young, 1984, p. 132]).

To reduce this correlation, as initially suggested in [Young, 1984, p. 132], the estimates can be filtered. The DT filter, denoted here by  $M_\theta(q^{-1})$  has the additional effect of smoothing the estimates, which improves the stability of the algorithm [Ljung and Söderström, 1983, p. 315]. A scheme of RSRIVC, including the filter  $M_\theta(q^{-1})$  and the stability test described previously, is shown in Figure 3.2. The filter  $M_\theta(q^{-1})$  can be an infinite impulse response (IIR) or a finite impulse response (FIR) filter.

**Suggested choice/procedure 3.5** *As proposed in [Wong and Polak, 1967], to reduce correlation between filtered instruments and filtered noise, we consider as default option for  $M_\theta(q^{-1})$  a delay, i.e.<sup>2</sup>*

$$M_\theta(q^{-1}) = q^{-1}$$

An alternative to filtering the estimates is to use regularization. This is discussed in Section 3.6.2.

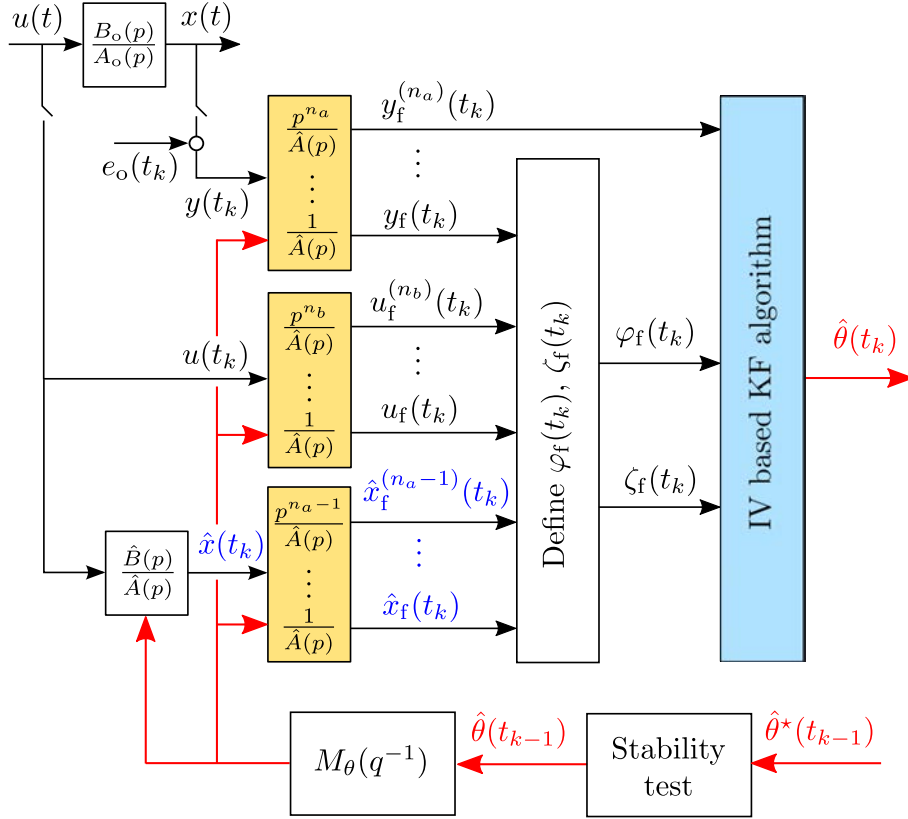
## 3.3 Estimation of the normalized covariance matrix $Q_n$

To apply Kalman filter based methods, the user has to specify the normalized covariance matrix  $Q_n$ . The performance of these methods, in terms of tracking ability and noise sensitivity, depends on  $Q_n$ . The maximum likelihood method is one of the available approaches that can be used to estimate this hyperparameter  $Q_n$  [Young, 2011].

### 3.3.1 Review of maximum likelihood method

The maximum likelihood method is presented in detail in [Åström, 1980] (see also [Söderström and Stoica, 1989, Ljung, 1999]). Here we recall this method considering a simple

<sup>2</sup>Unless otherwise stated, we consider  $M_\theta(q^{-1}) = q^{-1}$  both in the previous and the following examples.


 Figure 3.2: Diagram of RSRIVC including a DT filter  $M_\theta(q^{-1})$  and a stability test.

case. We assume that the system is modeled by a random walk for the parameters and a linear regression, which could represent a LTV ARX model, either in CT or DT, *i.e.*

$$\mathcal{M} \begin{cases} \theta(t_k) &= \theta(t_{k-1}) + w(t_k) \\ z(t_k) &= \psi^T(t_k)\theta(t_k) + e(t_k) \end{cases} \quad (3.12)$$

where  $w(t_k)$  and  $e(t_k)$  are independent zero-mean DT Gaussian noise processes with covariance matrix  $Q_w$  variance  $\sigma_e^2$ , respectively. For the estimation of (3.12), there is available  $N$  measurements of the regressor and the output denoted by  $\Psi_N = \{\psi(t_k)\}_{k=1}^N$  and  $Z_N = \{z(t_k)\}_{k=1}^N$ , respectively. Assuming that  $Z_N$  is a random sequence whose observations are not independent, the likelihood function, *i.e.* the probability density function of the observations  $Z_N$  given by the parameters  $\theta$  is defined by

$$\begin{aligned} p_\theta(Z_N|\Psi_N) &= p(Z_N|\Psi_N, \theta) \\ &= \prod_{k=1}^N p(z(t_k)|\Psi_k, \theta) \end{aligned} \quad (3.13)$$

The maximum likelihood method consists in finding the parameters  $\theta$  such that the measurements  $Z_N$  are as likely as possible [Ljung, 1999]. Mathematically, this is expressed as follows

$$\hat{\theta} = \arg \max_{\theta} p_\theta(Z_N|\Psi_N) \quad (3.14)$$

Due to numerical reasons, the logarithm of the likelihood, called log-likelihood, is usually maximized instead. Then, the maximization of the log-likelihood, denoted by  $l(\theta)$ , yields

the maximum likelihood estimates, *i.e.*

$$\begin{aligned}\hat{\theta} &= \arg \max_{\theta} \log p_{\theta}(Z_N | \Psi_N) \\ &= \arg \max_{\theta} l(\theta)\end{aligned}\tag{3.15}$$

If it is assumed that  $e(t_k)$  is a zero-mean Gaussian noise sequence with variance  $\sigma_e^2$ , then, the conditional densities in (3.13) are Gaussian, and the likelihood is given by

$$p_{\theta}(Z_N | \Psi_N) = \prod_{k=1}^N \frac{1}{\sqrt{2\pi r(t_k)}} \exp \left\{ -\frac{[z(t_k) - \hat{z}(t_k)]^2}{2r(t_k)} \right\}\tag{3.16}$$

with conditional mean  $\hat{z}(t_k)$  and conditional variance  $r(t_k)$  that are computed with the Kalman filter, *i.e.*

$$\begin{aligned}\hat{z}(t_k) &= \bar{\mathbb{E}}[z(t_k) | \Psi_k, \theta] \\ &= \psi^T(t_k) \hat{\theta}(t_k | t_{k-1})\end{aligned}\tag{3.17}$$

$$\begin{aligned}r(t_k) &= \text{var}[z(t_k) | \Psi_k, \theta] \\ &= \sigma_e^2 + \psi^T(t_k) P^*(t_k | t_{k-1}) \psi(t_k)\end{aligned}\tag{3.18}$$

Notice that in (3.18) the standard form of the Kalman filter is used, and  $P^*(t_k | t_{k-1}) / \sigma_e^2 = P(t_k | t_{k-1})$ . Then, the log-likelihood is given by

$$l(\theta) = -\frac{1}{2} \left\{ N \log 2\pi + \sum_{k=1}^N \log r(t_k) + \sum_{k=1}^N \frac{[z(t_k) - \hat{z}(t_k)]^2}{r(t_k)} \right\}\tag{3.19}$$

Finally, the maximization of (3.19) is equivalent to

$$\hat{\theta} = \arg \min_{\theta} \sum_{k=1}^N \log r(t_k) + \sum_{k=1}^N \frac{[z(t_k) - \hat{z}(t_k)]^2}{r(t_k)}\tag{3.20}$$

### 3.3.2 Estimation of $Q_n$ using maximum likelihood

In this section we will show how to estimate the normalized covariance matrix in a recursive off-line fashion considering the CT LTV OE model (2.28). This model can be written as the filtered linear regression (2.24) plus the random walk model for the parameters, *i.e.*

$$\mathcal{M} \begin{cases} \theta(t_k) = \theta(t_{k-1}) + w(t_k) \\ y_f^{(na)}(t_k) = \varphi_f^T(t_k) \theta(t_k) + v_f(t_k) \end{cases}\tag{3.21}$$

Notice that (3.21) is similar to (3.12). The difference is that the filtered noise  $v_f(t_k)$  in (3.21) is in general not Gaussian, although  $e(t_k)$  in (2.28) is assumed to be Gaussian. However, when RSRIVC is used,  $v_f(t_k)$  will be approximately asymptotically Gaussian. Thus, considering that the estimation is performed with RSRIVC, we assume that  $v_f(t_k)$  is Gaussian and the maximum likelihood method presented in the previous section can be applied to estimate the normalized covariance matrix  $Q_n$  of the Kalman filter. We will consider that  $Q_n$  is diagonal and its elements are denoted by  $q_{n,i}$  with  $i = 1, \dots, d$ . Therefore,  $Q_n$  is estimated by solving an optimization problem similar to (3.20) but with a parameter vector  $\eta$  that includes the constant scalars  $q_{n,i}$ , *i.e.*

$$\eta = [q_{n,1} \quad \dots \quad q_{n,d}]\tag{3.22}$$

Then,

$$\begin{aligned}\hat{\eta} &= \arg \min_{\eta} V \\ &= \arg \min_{\eta} \sum_{k=1}^N \log r(t_k) + \sum_{k=1}^N \frac{[z(t_k) - \hat{z}(t_k)]^2}{r(t_k)}\end{aligned}\quad (3.23)$$

The conditional mean

$$\hat{z}(t_k) = \hat{y}_f^{(n_a)}(t_k) = \varphi_f^T(t_k) \hat{\theta}(t_k|t_{k-1}) \quad (3.24)$$

and conditional variance

$$r(t_k) = \sigma_e^2 + \varphi_f^T(t_k) P^*(t_k|t_{k-1}) \varphi_f(t_k) \quad (3.25)$$

are computed with RSRIVC. The procedure requires to set initial values for  $Q_n$ , that should be defined considering Remark 1.3. We should also keep in mind that the user has to specify  $P(t_0)$ ,  $\hat{\theta}(t_0)$  and  $\lambda_{svf}$ . Additionally it might be necessary to incorporate bounds  $(l_{b,i}, u_{b,i})$  for  $q_{n,i}$ , as suggested in [Bavdekar et al., 2011] for a similar problem.

**Suggested choice/procedure 3.6** *From a physical insight of the system, it might be known that some parameters are constant or that some parameters vary at the same rate. To simplify the estimation of  $Q_n$  through maximum likelihood, it is important to use that prior knowledge. On the other hand, since we assume that the parameters are slowly varying, it is possible to identify local LTI models on segments of some training data. Then, from the set of estimated LTI model, we can learn about the parameter variations.*

### 3.3.3 Numerical example

For the following example, the normalized covariance matrix is estimated using maximum likelihood. The data generating system is given by

$$\mathcal{S} \begin{cases} (p + a_1^o(t))x(t) = b_0^o u(t) \\ y(t_k) = x(t_k) + e(t_k) \end{cases} \quad (3.26)$$

with  $b_0^o = 3$  and

$$a_1^o(t) = 2 + 0.8t/250$$

The sampling time is set to 0.025 s and the total simulation time is 250 s. The input is a PRBS and the DT measurement noise is a zero-mean, Gaussian noise with constant variance  $3.0 \cdot 10^{-3}$ . Part of the input-output data is shown in Figure 3.3.

It is assumed that the parameter  $b_0$  is known to be constant, and thus  $q_{n,2} = 0$ . Therefore,  $\eta$  is the scalar  $q_{n,1}$  corresponding to  $a_1$ . The optimization problem (3.23), with initial value  $q_{n,1} = 10^{-5}$ , lower bound  $l_b = 10^{-6}$ , upper bound  $u_b = 10^{-4}$  and  $\lambda_{svf} = 3$  (rad/s), yields  $q_{n,1} = 3.3 \cdot 10^{-5}$ . The plot cost function  $V$  vs  $\eta$  is shown in Figure 3.4.

The normalized covariance matrix is then given by

$$Q_n = \text{diag}\{3.3 \cdot 10^{-5} \quad 0\} \quad (3.27)$$

The true parameters are compared with the RSRIVC estimates computed with (3.27) in Figure 3.5.

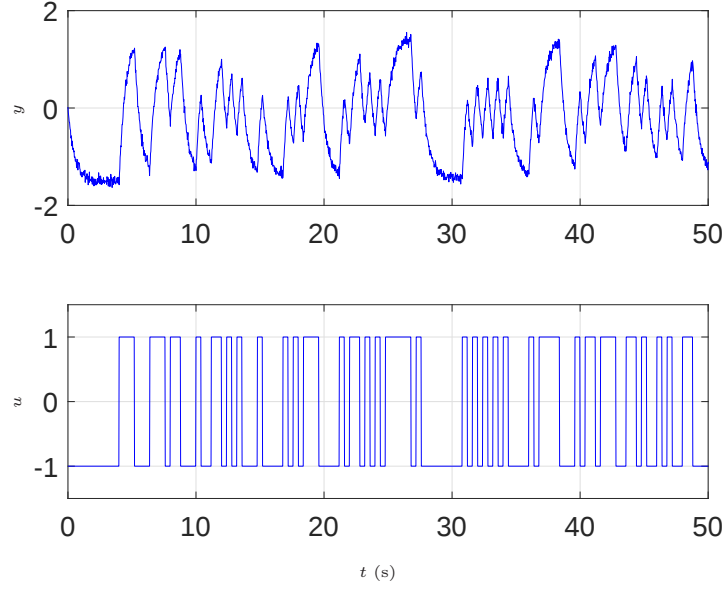


Figure 3.3: Part of the input-output data.

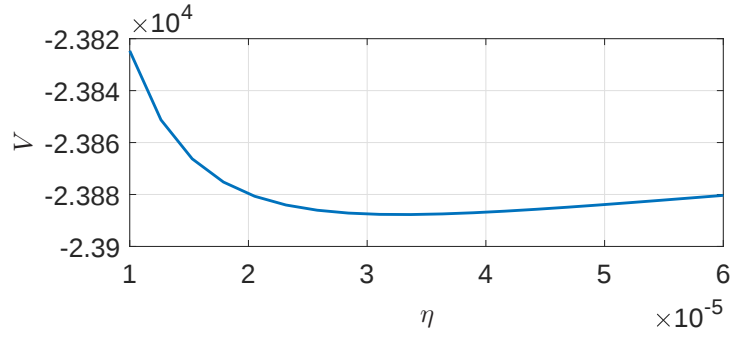


Figure 3.4: Plot of the cost function  $V$  vs  $\eta$ .

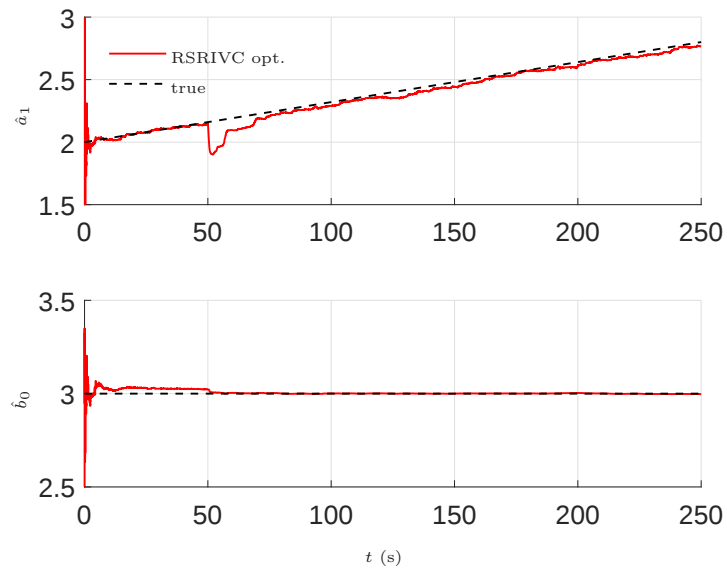


Figure 3.5: True parameters and RSRIVC estimates considering  $Q_n$  estimated by maximum likelihood.

### 3.4 Handling of input/output initial conditions

In the identification of DT models, the predictor depends on unknown input/output initial conditions. To circumvent that problem, the initial conditions can be assumed to be zero [Ljung, 1999, p. 320]. Analogously, in the derivation of the LSSVF method in Section 1.2.2, the initial conditions of the LTI system (1.1) have been ignored. That is, (1.22) has been obtained assuming that the initial conditions are zero. These initial conditions can be estimated together with the model parameters, as it was shown in [Garnier et al., 2004]. The goal of this section is to illustrate the impact of the initial conditions in the recursive estimates. To this end, we derive the RLSSVF for the LTI case considering initial conditions.

#### 3.4.1 Problem formulation

We start by formulating the problem again. Let us simplify the problem by assuming that there is no measurement noise in the data generating system, which is given by

$$\mathcal{S}: \quad A_o(p)y(t) = B_o(p)u(t) \quad (3.28)$$

with polynomials  $A_o(p)$  and  $B_o(p)$  relatively coprime. The system is assumed to be subject to an arbitrary set of input/output initial conditions

$$u_0 = \begin{bmatrix} u(0) & u^{(1)}(0) & \dots & u^{(n_b-1)}(0) \end{bmatrix}, \quad (3.29)$$

$$y_0 = \begin{bmatrix} y(0) & y^{(1)}(0) & \dots & y^{(n_a-1)}(0) \end{bmatrix}. \quad (3.30)$$

Moreover, it is assumed that the system belongs to the model set defined by

$$\mathcal{M}: \quad A(p, \theta)y(t) = B(p, \theta)u(t) \quad (3.31)$$

$A(p, \theta)$  and  $B(p, \theta)$  are the following polynomials

$$\begin{aligned} B(p, \theta) &= b_0 p^{n_b} + b_1 p^{n_b-1} + \dots + b_{n_b} \\ A(p, \theta) &= p^{n_a} + a_1 p^{n_a-1} + \dots + a_{n_a} \end{aligned}$$

where  $n_a \geq n_b$ . Then, the identification problem is to recursively estimate the parameters that characterize the model structure given by (3.31), based on the sequences  $\{u(t_k), y(t_k)\}_{k=1}^{N'}$ , where  $N'$  is the number of samples which increases by one with every recursion.

#### 3.4.2 Recursive least squares state-variable filter method

Consider the Laplace transform of the differential equation defined in (3.31),

$$s^{n_a}Y(s) + \sum_{i=1}^{n_a} a_i s^{n_a-i}Y(s) = \sum_{i=0}^{n_b} b_i s^{n_b-i}U(s) + \sum_{i=0}^{n_a-1} c_i s^{n_a-i} \quad (3.32)$$

or

$$A(s)Y(s) = B(s)U(s) + C(s) \quad (3.33)$$

where  $s$  represents the Laplace variable while  $Y(s)$  and  $U(s)$  are the Laplace transforms of  $y(t)$  and  $u(t)$  respectively. The coefficients  $c_i$  depend on the unknown parameters  $a_i$  and  $b_i$  as well as the unknown initial conditions. Notice that if there was measurement noise, the coefficients  $c_i$  would depend on initial conditions of the noise as well.

Applying a filter  $F(s)$  to (3.33) yields

$$A(s)F(s)Y(s) = B(s)F(s)U(s) + F(s)C(s), \quad (3.34)$$

The SVF has the following form

$$F(s) = \left( \frac{\lambda_{\text{svf}}}{s + \lambda_{\text{svf}}} \right)^{n_a} \quad (3.35)$$

where  $\lambda_{\text{svf}}$  is the cut-off frequency of the filter.

Let us define a filter  $F_i(s)$  to be

$$F_i(s) = \frac{(\lambda_{\text{svf}})^{n_a} s^i}{(s + \lambda_{\text{svf}})^{n_a}} \quad \text{for } i = 1, \dots, n_a \quad (3.36)$$

Then, (3.34) in terms of  $F_i(s)$  takes the following form

$$\begin{aligned} & (F_{n_a}(s) + a_1 F_{n_a-1}(s) + \dots + a_{n_a} F_0(s))Y(s) = \\ & (b_0 F_{n_b}(s) + \dots + b_{n_b} F_0(s))U(s) + (c_1 F_{n_a}(s) + \dots + c_{n_a} F_0(s)) \end{aligned} \quad (3.37)$$

In terms of time-domain signals, (3.37) can be written as

$$\begin{aligned} & [F_{n_a}y](t) + a_1[F_{n_a-1}y](t) + \dots + a_{n_a}[F_0y](t) = \\ & b_0[F_{n_b}u](t) + \dots + b_{n_b}[F_0u](t) + c_1 f_{n_a-1}(t) + \dots + c_{n_a} f_0(t) \end{aligned} \quad (3.38)$$

where

$$\begin{aligned} [F_i y](t) &= y_f^{(i)}(t) = f_i(t) * y(t) \\ [F_i u](t) &= u_f^{(i)}(t) = f_i(t) * u(t) \end{aligned}$$

with  $f_i$  the corresponding impulse response of (3.35) and  $*$  denoting the convolution operator. The filter outputs  $[F_i y]$  and  $[F_i u]$  are prefiltered time-derivatives of the outputs and inputs in the bandwidth of interest. Then, (3.38) can be expressed as the linear regression

$$y_f^{(n_a)}(t_k) = \varphi_f^T(t_k) \theta \quad (3.39)$$

The parameter vector is given by

$$\theta = [a_1 \quad \dots \quad a_{n_a} \quad b_0 \quad \dots \quad b_{n_b} \quad c_1 \quad \dots \quad c_{n_a}]^T \quad (3.40)$$

and the regressor by

$$\varphi_f(t_k) = \begin{pmatrix} \varphi_{f,y}(t_k) \\ \varphi_{f,u}(t_k) \\ \varphi_{ic}(t_k) \end{pmatrix} \quad (3.41)$$

with

$$\begin{aligned} \varphi_{f,y}^T(t_k) &= - \begin{pmatrix} y_f^{(n_a-1)}(t_k) & y_f^{(n_a-2)}(t_k) & \dots & y_f(t_k) \end{pmatrix} \\ \varphi_{f,u}^T(t_k) &= \begin{pmatrix} u_f^{(n_b)}(t_k) & u_f^{(n_b-1)}(t_k) & \dots & u_f(t_k) \end{pmatrix} \\ \varphi_{ic}^T(t_k) &= \begin{pmatrix} f_{n_a-1}(t_k) & f_{n_a-2}(t_k) & \dots & f_0(t_k) \end{pmatrix} \end{aligned}$$

From (3.39) it is straightforward to estimate  $\theta$  using the RLSSVF-FF algorithm (1.57) with  $\lambda = 1$  (for the LTI case considered here).

The estimation of initial conditions implies an extra computational burden. On the other hand, neglecting its effect can lead to overshoots in the estimates. In order to avoid these overshoots, another option is to start the estimation after transients due to initial conditions have died out.

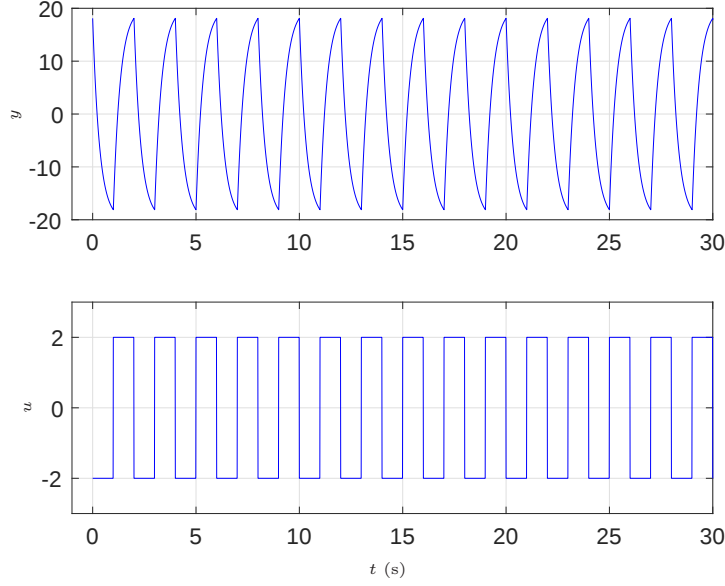


Figure 3.6: Part of the input-output data.

**Suggested choice/procedure 3.7** *The proposed identification strategy with RLSSVF is the following: prefiltered time-derivatives are computed from  $t = 0$ , but estimates are computed only from the time instant when the effect of the initial conditions are negligible. This time instant when the least squares approach starts to operate is denoted by  $t_{s,ls}$ . In the following example, these ideas are illustrated and further discussed.*

**Remark 3.2** *An alternative to deal with overshoots is to use regularization based methods, as pointed out e.g. in [Stoica and Åhgren, 2002]. The use of regularization is discussed in Section 3.6.*

### 3.4.3 Numerical example

The data generating system is noise-free and given by

$$(p + a_1^o)y(t) = b_0^o u(t) \quad (3.42)$$

with  $a_1^o = 3$  and  $b_0^o = 30$ . The sampling time is  $T_s = 0.01$  s and the input is a square wave. In the input-output data shown in Figure 3.6, the transient response has been removed. Therefore, the output is periodic for  $t \geq 0$  and the period is 2 s. The initial condition is  $y(0) = 18.1$ .

We are interested in identifying a CT model that represents (3.42) considering initial conditions. For this case, we have a linear regression as in (3.39) with

$$\begin{aligned} \varphi_f^T(t_k) &= \begin{bmatrix} -y_f(t_k) & u_f(t_k) & f_0(t_k) \end{bmatrix} \\ \theta^T &= \begin{bmatrix} a_1 & b_0 & c_1 \end{bmatrix} \end{aligned}$$

where  $f_0(t_k)$  is the impulse response corresponding to the SVF

$$F_0(s) = \frac{\lambda_{svf}}{s + \lambda_{svf}} \quad (3.43)$$

Notice that  $c_1$  is here an estimate of  $y(0)$ .

The parameters are estimated with RLSSVF using  $\lambda_{svf} = 4$  rad/s. In Figure 3.7, we compare RLSSVF estimates obtained with and without taking into account the initial

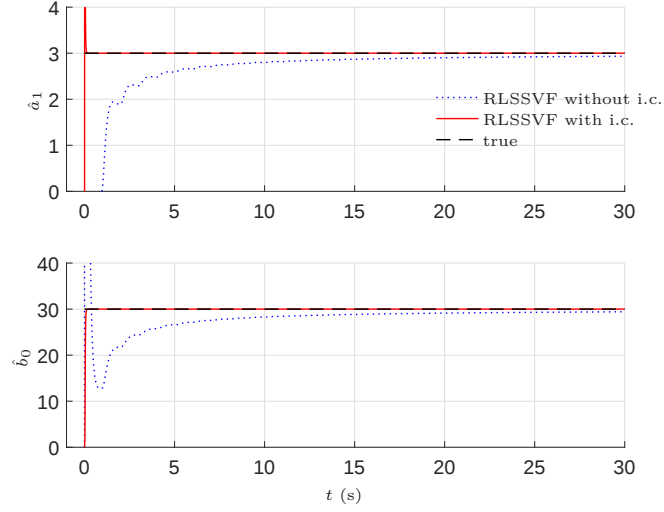


Figure 3.7: True parameters and RLSSVF estimates with and without initial condition handling.

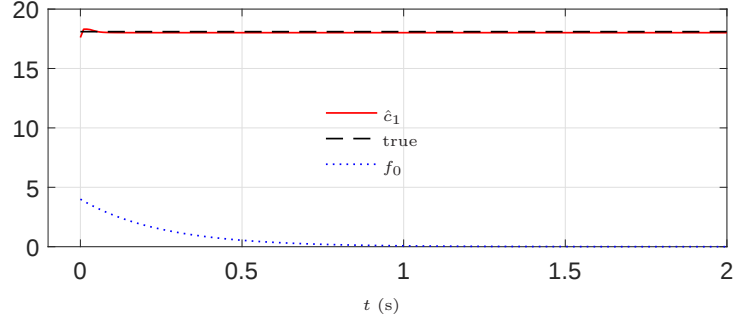


Figure 3.8: True parameter  $c_1$ , RLSSVF estimate (with i.c.)  $\hat{c}_1$  and impulse response  $f_0$ .

conditions. We can see that by taking into account the initial conditions the results are better in terms of convergence rate and bias. The estimate  $\hat{c}_1(t_k)$  is very close to the true value ( $y(0) = 18.1$ ), as it can be seen in Figure 3.8, where we plot additionally the impulse response  $f_0(t_k)$ .

Now let us take a look at the overshoots in the estimates that cannot be seen in Figure 3.7 because of the chosen axes. The overshoots are defined as follows

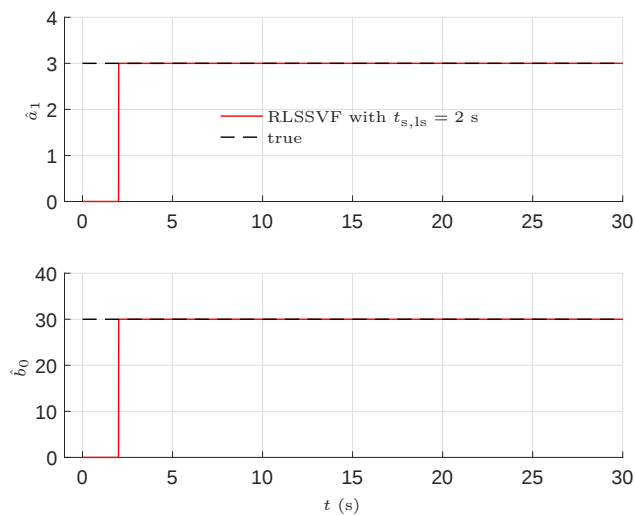
$$\text{overshoot} = \frac{\max[\hat{\theta}(t_k)] - \theta_o}{\theta_o} \times 100 \quad (3.44)$$

The results, presented in Table 3.1, show that large overshoots are obtained when the initial conditions are not considered. By including the estimate of  $c_1$ , the overshoots are significantly reduced.

As it was mentioned in the previous section, another option is to start the estimation after transients due to initial conditions have died out. From Figure 3.8, for the chosen value of  $\lambda_{\text{svf}}$ , we can see that the effect of the initial condition is negligible for  $t \geq 1$  s. The RLSSVF estimates with  $t_{\text{s,ls}} = 2$  s is shown in Figure 3.9. Note that since the input-output data is periodic and there is no noise, the observations for  $t = 2$  s are the same than for  $t = 0$ . These simulation conditions have been deliberately chosen in order to show only the effect of the initial conditions. In Table 3.1, we can see that the overshoots are the smallest in this case.

Table 3.1: Overshoots for the estimates  $\hat{a}_1$  and  $\hat{b}_0$ .

| Method & settings            | overshoots (%) for: |             |
|------------------------------|---------------------|-------------|
|                              | $\hat{a}_1$         | $\hat{b}_0$ |
| RLSSVF without i.c.          | 6561                | 5177        |
| RLSSVF with i.c.             | 110                 | 0.005       |
| RLSSVF with $t_{s,ls} = 2$ s | 0.0052              | 0.0193      |


Figure 3.9: True parameters and RLSSVF estimates using  $t_{s,ls} = 2$  s.

From (3.43) it can be seen that the effect of the initial conditions is weighted by  $\lambda_{\text{svf}}$ . Therefore, if RLSSVF estimates are computed without initial conditions, the larger the value of  $\lambda_{\text{svf}}$ , the larger the overshoots will be. This has been also pointed out in [Johansson, 1994], for an approach which is similar to RLSSVF.

### 3.5 Numerical issues

In off-line identification, least squares estimates can be computed using the normal equations. Nonetheless, the conditioning can be improved by using QR factorization. Another complementary method to improve conditioning is signal scaling.

While off-line estimation is usually done in a desktop computer, on-line estimation is meant to be performed in a microcontroller, where numerical aspects are more important. Roundoff errors introduced by the microcontroller due to its poor machine precision might yield a negative definite covariance matrix. In recursive estimation, QR factorization can be also implemented in a recursive fashion with forgetting factor (see [Bittencourt et al., 2015]). Thus, the forgetting factor based RLSSVF method (1.57) can be implemented using QR factorization. However, the Kalman filter based RLSSVF cannot be handled in this way. A solution is Potter's algorithm which corresponds to a square root filtering method (see *e.g.* [Niedźwiecki, 2000, Grewal and Andrews, 2015, Simon, 2006]). For Potter's algorithm, there are both a Kalman filter based approach and a forgetting factor based approach.

Next, we discuss first the numerical issues in off-line identification. After showing how to measure conditioning in LSSVF, we present the signal scaling approach for LSSVF to improve conditioning. Then, the numerical issues in recursive identification are addressed. First we discuss solutions for LS based methods and at the end for the IV based methods developed in Chapter 2. In Section 4.1, the proposed approaches to improve conditioning are tested with real data.

#### 3.5.1 Conditioning in LSSVF

For LSSVF presented in Section 1.2.2, let us consider the minimization problem (1.10) in matrix form. Then, the cost function is given by

$$V = \frac{1}{N} \|Y - \Phi\theta\|^2 \quad (3.45)$$

where

$$Y = \begin{pmatrix} y_{\text{f}}^{(n_a)}(t_1) \\ y_{\text{f}}^{(n_a)}(t_2) \\ \vdots \\ y_{\text{f}}^{(n_a)}(t_N) \end{pmatrix}, \quad \Phi = \begin{pmatrix} \Phi_y & \Phi_u \end{pmatrix}$$

with

$$\Phi_y = - \begin{pmatrix} y_f^{(n_a-1)}(t_1) & y_f^{(n_a-2)}(t_1) & \cdots & y_f(t_1) \\ y_f^{(n_a-1)}(t_2) & y_f^{(n_a-2)}(t_2) & & \vdots \\ \vdots & \vdots & & \vdots \\ y_f^{(n_a-1)}(t_N) & y_f^{(n_a-2)}(t_N) & \cdots & y_f(t_N) \end{pmatrix}$$

$$\Phi_u = \begin{pmatrix} u_f^{(n_b)}(t_1) & u_f^{(n_b-1)}(t_1) & \cdots & u_f(t_1) \\ u_f^{(n_b)}(t_2) & u_f^{(n_b-1)}(t_2) & & \vdots \\ \vdots & \vdots & & \vdots \\ u_f^{(n_b)}(t_N) & u_f^{(n_b-1)}(t_N) & \cdots & u_f(t_N) \end{pmatrix}$$

The optimization problem with objective function (1.10) corresponds to the normal equations

$$[\Phi^T \Phi] \hat{\theta} = [\Phi^T Y] \quad (3.46)$$

or

$$R_N \hat{\theta} = f_N \quad (3.47)$$

which have a closed-form solution

$$\hat{\theta} = R_N^{-1} \cdot f_N \quad (3.48)$$

There are different ways to check whether  $R_N$  in (3.48) is well-conditioned. If  $R_N$  is well-conditioned, it is invertible. The invertibility of a matrix can be evaluated by computing its determinant which must be different than zero. However, this might not be sufficient to detect numerically ill-conditioned matrices whose determinant might be close to zero [Shardt and Huang, 2013]. An alternative is to use the condition number, which is defined by [Quarteroni et al., 2000, p. 60]

$$\kappa_p(R_N) = \|R_N\|_p \|R_N^{-1}\|_p \quad (3.49)$$

where  $\|\cdot\|_p$  is the  $p$ -norm. For the infinity norm ( $p = \infty$ ), the values tend to be too conservative, and usually 2-norm ( $p = 2$ ) is preferred [Shardt and Huang, 2013]. The 2-norm condition number is defined by

$$\kappa_2 = \frac{\sigma_{\max}}{\sigma_{\min}} \quad (3.50)$$

where  $\sigma_{\max}$ ,  $\sigma_{\min}$  is the maximum and minimum singular values, respectively. For a symmetric positive definite matrix (like in this case for  $R_N$ )

$$\kappa_2 = \frac{\lambda_{\max}}{\lambda_{\min}} \quad (3.51)$$

with  $\lambda_{\max}$ ,  $\lambda_{\min}$  the maximum and minimum eigenvalues, respectively [Quarteroni et al., 2000, p. 61].

**Remark 3.3** *The condition number  $\kappa_2(M)$  lies in the interval  $[1, \infty)$  and the optimal conditioning is 1, which is obtained in DT identification with a FIR model with white noise as input [Van den Hof and Ninness, 2005].*

### 3.5.2 Signal scaling in LSSVF

Signal scaling can be used to improve conditioning of the normal equations. In the off-line identification of a DT linear regression model using least squares, the regressor is defined in term of past outputs and inputs (see (1.5)). Thus, the linear regression can be readily scaled by changing the magnitude of the input-output signals. The situation is more tricky in the off-line identification of a CT linear regression model using LSSVF. In that case, we arrive at a filtered linear regression (see (1.24)) whose regressor is defined in terms of prefiltered time-derivatives of the input-output signals. Thus, the filtered linear regression cannot be longer scaled by changing the magnitude of the input-output signals.

Next we show how to apply signal scaling in LSSVF. The cost function (3.45) can be modified as follows

$$V = \frac{1}{N} \|Y - \Phi M_s M_s^{-1} \theta\|^2 \quad (3.52)$$

where  $M_s$  is a diagonal matrix that is used to obtain a scaled solution. The matrix  $M_s$  is defined by

$$M_s = \begin{bmatrix} m_{s,1} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & m_{s,n_\theta} \end{bmatrix} \quad (3.53)$$

with

$$m_{s,i} = \frac{1}{\|\Phi_i\|} \quad (3.54)$$

and  $\Phi_i$  the column vectors that constitute  $\Phi$ , i.e.

$$\Phi = \begin{bmatrix} \Phi_1 & \dots & \Phi_{n_\theta} \end{bmatrix} \quad (3.55)$$

Let us define

$$\hat{\theta}_s = M_s^{-1} \hat{\theta} \quad (3.56a)$$

$$\Phi_s = \Phi M_s \quad (3.56b)$$

Then, the cost function (3.52) may be written as

$$V = \frac{1}{N} \|Y - \Phi_s \theta_s\|^2 \quad (3.57)$$

Thus, the scaled LSSVF estimate that minimizes (3.57) is given by,

$$\hat{\theta}_s = [\Phi_s^T \Phi_s]^{-1} \Phi_s^T Y \quad (3.58)$$

In general, the filtered noise  $v_f(t_k)$  is not Gaussian. However, if the polynomial  $A(p, \theta)$  in 1.2 and the polynomial  $E(p)$  in (1.19) are equal,  $v_f(t_k)$  is Gaussian and the parameter covariance matrix can be estimated by

$$P = \hat{\sigma}_{v_f}^2 [\Phi^T \Phi]^{-1} \quad (3.59)$$

with  $\hat{\sigma}_{v_f}(t_k)$  the sample variance of  $v_f(t_k)$ . Equation (3.59), that will allow us to introduce the signal scaling approach in the recursive case in the next section, can be written as follows

$$P[\Phi^T \Phi] = \hat{\sigma}_{v_f}^2 \quad (3.60)$$

Replacing (3.56b) in (3.60) yields

$$P M_s^{-1} \Phi_s^T \Phi_s M_s^{-1} = \hat{\sigma}_{v_f}^2 \quad (3.61)$$

Multiplying (3.61) from the left by  $M_s^{-1}$  we obtain

$$M_s^{-1} P M_s^{-1} \Phi_s^T \Phi_s M_s^{-1} = \hat{\sigma}_{v_f}^2 M_s^{-1} \quad (3.62)$$

Let us define

$$P_s = M_s^{-1} P M_s^{-1} \quad (3.63)$$

Then, multiplying (3.62) from the right by  $M_s$  yields

$$P_s = \hat{\sigma}_{v_f}^2 [\Phi_s^T \Phi_s]^{-1} \quad (3.64)$$

Then, by choosing  $M_s$  according to (3.53),  $\kappa_2(P_s) < \kappa_2(P)$ . From the properties of the condition number, note that for  $R_N$  defined in (3.48),  $\kappa_2(R_N) = \kappa_2(P)$ .

### 3.5.3 Signal scaling in RLSSVF

Signal scaling can be also applied in recursive estimation. The expression (3.56a) shows us how to scale  $\hat{\theta}(t_k)$  in the KF based RLSSVF algorithm defined in (1.59). Note that although (3.63) is in general not valid, it also shows us how to modify  $P(t_k)$  in (1.59). Then, in analogy to the off-line case (see (3.56a) and (3.63)), for the time instant  $t_k$ , we can write

$$\hat{\theta}_s(t_k) = M_s^{-1} \hat{\theta}(t_k) \quad (3.65a)$$

$$P_s(t_k) = M_s^{-1} P(t_k) M_s^{-1} \quad (3.65b)$$

Replacing (3.65) in (1.59) we obtain the scaled version of the KF based RLSSVF:

*Prediction step:*

$$\hat{\theta}_s(t_k|t_{k-1}) = \hat{\theta}_s(t_{k-1}) \quad (3.66a)$$

$$P_s(t_k|t_{k-1}) = P_s(t_{k-1}) + Q_{n,s} \quad (3.66b)$$

*Correction step:*

$$\hat{\theta}_s(t_k) = \hat{\theta}_s(t_k|t_{k-1}) + M_s^{-1} L(t_k) \varepsilon(t_k) \quad (3.66c)$$

$$\varepsilon(t_k) = y_f^{(n_a)}(t_k) - \varphi_f^T(t_k) M_s \hat{\theta}_s(t_k|t_{k-1}) \quad (3.66d)$$

$$L(t_k) = \frac{M_s P_s(t_k|t_{k-1}) M_s \varphi_f(t_k)}{1 + \varphi_f^T(t_k) M_s P_s(t_k|t_{k-1}) M_s \varphi_f(t_k)} \quad (3.66e)$$

$$P_s(t_k) = M_s^{-1} [I - L(t_k) \varphi_f^T(t_k)] M_s P_s(t_k|t_{k-1}) \quad (3.66f)$$

where  $Q_{n,s} = M_s^{-1} Q_n M_s^{-1}$ . Notice that in this recursive version  $M_s$  could be computed with some representative training data. It is clear from (3.66) that the disadvantage of this approach are the additional operations that are introduced.

### 3.5.4 Review of Potter's algorithm

Next we recall Potter's algorithms which have been developed for DT estimation. Since the only difference in the CT case w.r.t. the DT case is how the linear regression is built, we present directly the CT version of the approaches.

The conditioning of the recursive estimation can be also improved by using square root filtering algorithms (see *e.g.* [Grewal and Andrews, 2015]). Instead of propagating the covariance matrix  $P(t_k)$ , they propagate  $S(t_k)$  which is the square root of  $P(t_k)$ , i.e.

$$P(t_k) = S(t_k) S^T(t_k) \quad (3.67)$$

The benefit of this approach is clear, since the condition number of  $S(t_k)$  is the square root of the condition number of  $P(t_k)$ , i.e.

$$\kappa_2(P(t_k)) = \kappa_2^2(S(t_k)) \quad (3.68)$$

One of these approaches is Potter's algorithm, which consists on writing the covariance matrix of the correction step as follows

$$S(t_k)S^T(t_k) = S(t_{k-1}) \left[ I_d - \frac{f(t_k)f^T(t_k)}{\beta(t_k)} \right] S(t_{k-1}) \quad (3.69)$$

where

$$f(t_k) = S^T(t_{k-1})\varphi_f(t_k) \quad (3.70a)$$

$$\beta(t_k) = 1 + f^T(t_k)f(t_k) \quad (3.70b)$$

It has been shown that the square root of  $P(t_k)$  can be updated as follows (see *e.g.* [Niedźwiecki, 2000])

$$S(t_k) = S(t_{k-1}) \left[ I_d - \frac{f(t_k)f^T(t_k)}{\alpha(t_k)} \right] \quad (3.71)$$

where

$$\alpha(t_k) = \beta(t_k) + \sqrt{\beta(t_k)} \quad (3.72)$$

The Kalman filter based Potter's algorithm is given by

*Prediction step:*

$$\hat{\theta}(t_k|t_{k-1}) = \hat{\theta}(t_{k-1}) \quad (3.73a)$$

$$\begin{bmatrix} S^T(t_k|t_{k-1}) \\ 0 \end{bmatrix} = T \begin{bmatrix} S^T(t_{k-1}) \\ Q_n^{T/2} \end{bmatrix} \quad (3.73b)$$

*Correction step:*

$$\hat{\theta}(t_k) = \hat{\theta}(t_k|t_{k-1}) + L(t_k)\varepsilon(t_k) \quad (3.73c)$$

$$\varepsilon(t_k) = y_f^{(n_a)}(t_k) - \varphi_f^T(t_k)\hat{\theta}(t_k|t_{k-1}) \quad (3.73d)$$

$$f(t_k) = S^T(t_k|t_{k-1})\varphi_f(t_k) \quad (3.73e)$$

$$\beta(t_k) = 1 + f^T(t_k)f(t_k) \quad (3.73f)$$

$$\alpha(t_k) = \beta(t_k) + \sqrt{\beta(t_k)} \quad (3.73g)$$

$$L(t_k) = \frac{S(t_k|t_{k-1})f(t_k)}{\beta(t_k)} \quad (3.73h)$$

$$S(t_k) = S(t_k|t_{k-1}) \left[ I_d - \frac{f(t_k)f^T(t_k)}{\alpha(t_k)} \right] \quad (3.73i)$$

where  $T$  is a  $2n_\theta \times 2n_\theta$  orthogonal matrix that can be found using Givens rotations, Householder transformation, etc. (see more in [Simon, 2006, p. 162]). Regarding the matrix  $S(t_k|t_{k-1})$ , note that it is not unique. Algorithm (3.73) will be called PotterSVF. Signal scaling can also be introduced in PotterSVF as it is shown in Appendix C.

### 3.5.5 Signal scaling in RSRIVC

The RSRIVC method presented in Section 2.3.3 is the on-line counterpart of SRIVC. The SRIVC estimate can be written as in (3.48), *i.e.*

$$\hat{\theta}^i = \left[ \sum_{k=1}^N \zeta_f(t_k, \hat{\theta}^{i-1}) \varphi_f^T(t_k, \hat{\theta}^{i-1}) \right]^{-1} \cdot \left[ \sum_{k=1}^N \zeta_f(t_k, \hat{\theta}^{i-1}) y_f^{(n_a)}(t_k, \hat{\theta}^{i-1}) \right]$$

This approach corresponds to the so-called non-symmetric version of the IV technique [Söderström and Stoica, 1983], since in this case, the matrix in the first square brackets is non-symmetric. An alternative is the symmetric SRIVC defined by

$$\hat{\theta}^i = \left[ \sum_{k=1}^N \zeta_f(t_k, \hat{\theta}^{i-1}) \zeta_f^T(t_k, \hat{\theta}^{i-1}) \right]^{-1} \cdot \left[ \sum_{k=1}^N \zeta_f(t_k, \hat{\theta}^{i-1}) y_f^{(n_a)}(t_k, \hat{\theta}^{i-1}) \right]$$

where in this case, the matrix in the first square brackets is symmetric. Analogously, we can derive the recursive counterpart of the symmetric SRIVC.

The drawback of the square root filtering method is that it cannot be applied to non-symmetric IV techniques, since the corresponding covariance matrix is non-symmetric and it cannot be written as (3.67). Such an issue could be circumvented by using the symmetric IV approach. However, it is known that the non-symmetric IV approach gives better results than the symmetric IV approach [Söderström and Stoica, 1983]. A solution to improve conditioning in the proposed recursive IV methods presented in Chapter 2 is then to use signal scaling. Analogously to RLSSVF (see Section 3.5.3), the scaled version of the KF based RSRIVC method is given by

*Prediction step:*

$$\hat{\theta}_s(t_k|t_{k-1}) = \hat{\theta}_s(t_{k-1}) \quad (3.74a)$$

$$P_s(t_k|t_{k-1}) = P_s(t_{k-1}) + Q_{n,s} \quad (3.74b)$$

*Correction step:*

$$\hat{\theta}_s(t_k) = \hat{\theta}_s(t_k|t_{k-1}) + M_s^{-1} L(t_k) \varepsilon(t_k) \quad (3.74c)$$

$$\varepsilon(t_k) = y_f^{(n_a)}(t_k) - \varphi_f^T(t_k) M_s \hat{\theta}_s(t_k|t_{k-1}) \quad (3.74d)$$

$$L(t_k) = \frac{M_s P_s(t_k|t_{k-1}) M_s \zeta_f(t_k)}{1 + \varphi_f^T(t_k) M_s P_s(t_k|t_{k-1}) M_s \zeta_f(t_k)} \quad (3.74e)$$

$$P_s(t_k) = M_s^{-1} [I - L(t_k) \varphi_f^T(t_k)] M_s P_s(t_k|t_{k-1}) \quad (3.74f)$$

where  $Q_{n,s} = M_s^{-1} Q_n M_s^{-1}$ .

## 3.6 Anti-windup techniques

In system identification, in order to obtain accurate models, the input  $u$  of the system has to be persistently exciting (or sufficiently rich). Loosely speaking this means that the input should excite all modes of the system [Ljung and Söderström, 1983]. In formal terms, for off-line estimation using LSSVF, the input is persistently exciting if the matrix  $R_N$  in (3.48) is non-singular. Similarly, in on-line estimation using RLSSVF, the persistence of excitation condition is defined by

$$\sum_{k=i}^{i+s} \varphi_f(t_k) \varphi_f^T(t_k) \succeq c \cdot I \quad \forall k \quad (3.75)$$

with  $s$  and  $c$  positive constants. Notice that for stochastic regressors, the deterministic persistence of excitation condition (3.75) is very difficult to fulfill (see [Ljung and Gunnarsson, 1990] and [Niedźwiecki, 2000, p. 72]). In our case, if the input is deterministic like a PRBS, and the output is contaminated with stochastic noise, the regressor will have both deterministic and stochastic components. Then (3.75) could be used to check the persistency of excitation. For the recursive IV based methods, the counterpart of (3.75) is given by

$$\sum_{k=i}^{i+s} \zeta_f(t_k) \varphi_f^T(t_k) \succeq c \cdot I \quad \forall k \quad (3.76)$$

### 3.6.1 Anti-windup techniques in discrete-time identification

In DT recursive identification, when the input is not sufficiently rich, recursive estimation approaches could suffer from a phenomenon called estimator windup, where the covariance matrix blow up leading to wrong estimates. To illustrate the problem, let us consider an extreme case where we want to estimate a FIR model,

$$y(t_k) = \phi^T(t_k) \rho + e(t_k)$$

with  $\phi(t_k) = 0, \forall t_k \geq 0$ . Then, the RLS-FF algorithm becomes [Åström and Wittenmark, 2008, p. 473]

$$\hat{\rho}(t_k) = \hat{\rho}(t_{k-1}) \quad (3.77a)$$

$$P(t_k) = \frac{1}{\lambda} P(t_{k-1}) \quad (3.77b)$$

The equations (3.77) are unstable and  $P(t_k)$  will grow exponentially if  $\lambda < 1$ . On the other hand, the KF algorithm with  $\phi(t_k) = 0$  becomes [Niedźwiecki, 2000, p. 284], [Evestedt, 2007]

$$\hat{\rho}(t_k) = \hat{\rho}(t_{k-1}) \quad (3.78a)$$

$$P(t_k) = P(t_{k-1}) + Q_w(t_k) \quad (3.78b)$$

The equations (3.78) are also unstable but in this case  $P(t_k)$  grows linearly.

**Remark 3.4** *In general, it can be expected that a linear growth is slower than an exponential growth, which means that the Kalman filter with a constant  $Q_w$  could be more robust to excitation problems than the forgetting factor based approach [Cao and Schwartz, 2004].*

Many anti-windup techniques have been proposed in the literature (see *e.g.* [Salgado et al., 1988, Parkum et al., 1992, Gunnarsson, 1996, Cao and Schwartz, 2000, Evestedt, 2007, Evestedt et al., 2008]). We can distinguish for instance the following classes:

- Directional forgetting approaches. The principle is that old data must be discarded only in the direction where new information is available. One algorithm of this class is the selective forgetting which was developed in [Parkum, 1992, Parkum et al., 1992].
- Directional tracking approaches. The principle is to restrict tracking directions of an algorithm to the excited subspace (see [Cao and Schwartz, 2004]). The algorithm proposed in [Stenlund and Gustafsson, 2002] corresponds to this class. It is a KF based algorithm and it is suitable when the parameters vary at different speeds.

- Conditional updating [Åstrom and Wittenmark, 2008, p. 477] (to be explained in Section 3.6.2).
- Regularization based approaches [Gunnarsson, 1996] (to be explained in Section 3.6.3).

These methods have been developed for DT estimation. Here we extend the conditional updating and regularization based approaches to the CT case using the linear filter methods that we have proposed. Since the only difference in the CT case w.r.t. the DT case is how the linear regression is built, we present directly the CT version of the approaches.

### 3.6.2 Conditional updating

This method consists in updating the estimates and covariance only when there is excitation [Åstrom and Wittenmark, 2008, p. 477]. A method that works with conditional updating is for instance the Dasgupta-Huang optimal bounding ellipsoid algorithm [Dasgupta and Huang, 1987]. Note that conditional updating can be done using the same techniques proposed in *data quality assessment* [Peretzki et al., 2011, Shardt and Huang, 2013, Bittencourt et al., 2015], which consists on finding suitable data for model estimation given a large dataset.

There are different measures that can be used for conditional updating and the criteria to choose one or another should be based *e.g.* on computational cost, reliability and sensitivity of the indexes. By sensitivity we mean how much does the measure change between rich excitation and poor excitation.

The conditional updating can be based on:

1. the conditions of persistent excitation (3.75) and (3.76) for LS and IV based methods, respectively
2. measures of the input  $u(t_k)$  or output  $y(t_k)$
3. measures of quantities coming from the recursive algorithm like the scalars  $\varepsilon(t_k)$  and  $\varphi_f^T(t_k)P(t_k)\varphi_f(t_k)$  [Åstrom and Wittenmark, 2008], or the matrix  $P(t_k)$

Regarding the second category, the updating could be done for instance based on a variability test of the output. In the context of data quality assessment, this is proposed in [Bittencourt et al., 2015] by means of a recursive computation of the output variance.

Regarding the third category, the updating can be done by defining a scalar measure of the covariance matrix  $P(t_k)$ . The inverse of  $P(t_k)$  can be proportional to the information matrix. Then good estimates imply a small matrix  $P(t_k)$  or a large information matrix. The size of a matrix can be measured in different ways, as it is discussed in the field of optimal experiment design [Rojas, 2008]. Possible solutions are the determinant or the trace. The other option is to compute the condition number<sup>3</sup> of  $P(t_k)$ , as it has been proposed in the studies about data quality assessment (see *e.g.* [Bittencourt et al., 2015]). Since the condition number lies in the interval  $[1, \infty)$  it might be difficult to set a threshold for it. Instead, its inverse, *i.e.*  $[\kappa_2(P)]^{-1}$  that lies in the interval  $(0, 1]$ , can be monitored [Bittencourt et al., 2015].

Note that in practice it is likely that conditional updating is based on several measures, in order to increase robustness of the approach. In [Bittencourt et al., 2015] for instance, various indexes are considered for data quality assessment.

---

<sup>3</sup>The condition number of a matrix  $A$ , denoted by  $\kappa_2(A)$ , is defined in Section 3.5.1.

### Proposed conditional updating for IV based methods

In this work, we suggest to use the recursive IV methods proposed in Chapter 2 with conditional updating based on the condition number of  $P(t_k)$ . However, conditional updating based on measures provided by the algorithm, like the condition number of  $P(t_k)$ , should not be done using directly an IV method. The reason is that in this case,  $P(t_k)$  depends not only on the data but also indirectly on the estimated parameters  $\hat{\theta}(t_k)$  through the filtered instrument  $\zeta_f(t_k)$ . Under poor excitation,  $\hat{\theta}(t_k)$  is not reliable, and neither is  $P(t_k)$  as a measure for conditional updating. The solution that we propose is to run in parallel RLSSVF for the purpose of obtaining  $\kappa_2(P)$ .

For RLSSVF there are two possible adaptation mechanism, the forgetting factor and the KF. The former is a better option for conditional updating because it is less robust to poor excitation (see Remark 3.4), which means that the measure  $\kappa_2(P)$  is more sensitive. Then, the conditional updating for the IV methods is based on the condition

$$\kappa_2(P) < \gamma \quad (3.79)$$

where  $P$  is obtained from RLSSVF-FF and  $\gamma$  is the threshold which is chosen depending on the given identification problem.

When the excitation is detected as being poor such that (3.79) does not hold, the IV based algorithms must be reset to the original values, *i.e.* the parameter estimates and covariance matrix are set as follows

$$\begin{aligned} \hat{\theta}(t_k) &= \theta(t_0) \\ P(t_k) &= P(t_0) \end{aligned}$$

Once the excitation is again rich enough, the IV approaches are initialized as usual with RLSSVF.

### 3.6.3 Levenberg-Marquardt regularization method

Regularization is a well-known technique in system identification (see e.g. [Sjöberg et al., 1993, Pillonetto et al., 2014]). It is often used to deal with an ill-conditioned problem that can arise due to poor excitation. Anti-windup algorithms based on regularization are presented in [Gunnarsson, 1996, Yoo et al., 2003, Waterschoot et al., 2008]. It is important to keep in mind that regularization involves a bias-variance trade-off.

As it is shown in [Sayed and Kailath, 1994], the forgetting factor based RLSSVF algorithm (1.57), initialized with  $\theta_0 = \theta^*$ ,  $P_0 = \Gamma^{-1}$ , corresponds to the following regularized cost function

$$V = \sum_{k=1}^{N'} \lambda^{N'-k} [y_f^{(n_a)}(t_k) - \varphi_f^T(t_k)\theta]^2 + [\theta - \theta^*](\lambda^{(N'+1)}\Gamma)[\theta - \theta^*] \quad (3.80)$$

In (3.80) we can note that as the number of samples increases, the regularization term vanishes due to the forgetting factor. By using regularization as in (3.80) we are forcing our first estimates to be close to  $\theta^*$ .

**Remark 3.5** *To increase the convergence rates of the estimates in the transient mode, it is sometimes suggested to decrease the value of the forgetting factor (see [Ljung and Söderström, 1983, p. 279]). In this way, the tracking capability of the algorithm is increased, but at the same time it is more sensitive to noise. On the other hand, it can be observed from (3.80), that by decreasing the forgetting factor, we penalize less the regularization term, which is damping the estimates.*

If we want the regularization term to remain in time, then the cost function should be defined as follows

$$V = \sum_{k=1}^{N'} \lambda^{N'-k} [y_f^{(n_a)}(t_k) - \varphi_f^T(t_k)\theta]^2 + [\theta - \theta^*]^T \Gamma [\theta - \theta^*] \quad (3.81)$$

The recursive estimate that minimize (3.81) for  $\theta^* = \theta(t_{N'-1})$  is presented in [Waterschoot et al., 2008] and it is defined as follows

$$\hat{\theta}(t_k) = \hat{\theta}(t_{k-1}) + R^{-1}(t_k) \varphi_f(t_k) \varepsilon(t_k) \quad (3.82a)$$

$$R(t_k) = \lambda R(t_{k-1}) + \varphi_f(t_k) \varphi_f^T(t_k) + (1 - \lambda) \Gamma \quad (3.82b)$$

$$\varepsilon(t_k) = y_f^{(n_a)}(t_k) - \varphi_f^T(t_k) \hat{\theta}(t_{k-1}) \quad (3.82c)$$

For the choice  $\theta^* = \theta(t_{N'-1})$ , algorithm (3.82) corresponds to the Levenberg-Marquardt regularization (LMR) approach in [Ljung and Söderström, 1983]. The algorithm (3.82) is written in terms of the CT linear regression that it is built with SVF. Therefore, we call it LMR-RLSSVF method, and it has two hyperparameters,  $\lambda$  and  $\Gamma$ . Notice that with this choice for  $\theta^*$ , we are damping the estimate such that it remains close to the previous value. The downside of LMR is that it involves matrix inversion, as it is the case for all the methods based on regularization. However, different methods have been developed to overcome or alleviate that issue (see [Gunnarsson, 1996] and [Waterschoot et al., 2008]).

### LMR approach and IV methods under rich excitation

While regularization introduces bias in the estimates, the aim of IV methods is to reduce it. Therefore, at first sight, it would seem that it does not make sense to use these approaches together. However, in recursive estimation of slowly time-varying models under rich excitation conditions, if we consider regularization as in (3.81) with  $\theta^* = \theta(t_{N'-1})$ , the bias introduced is minimum. On the other hand, in the IV techniques presented in Chapter 2, the instruments are built using previous estimates, which should vary smoothly to guarantee stability of the algorithm. A solution to get smooth estimates is to use a filter, as discussed in Section 3.2.3. Another option is to use the IV methods with regularization, *i.e.* the IV counterpart of algorithm (3.82).

However, it is important to stress that such an approach is not suitable to deal with poor excitation, because in the considered IV methods the instruments depend on previous estimates  $\hat{\theta}(t_{k-1})$  that are not reliable.

#### 3.6.4 Numerical example

To illustrate the windup problem and the proposed solutions, the numerical example given in Section 2.3.5 is considered under poor excitation conditions. For  $0 \leq t < 6000$  s the input is a PRBS as before, and for  $6000 \leq t \leq 9000$  s the input remains equal to 1, leading to an estimator windup problem. Part of the input-output data is shown in Figure 3.10.

We present here the performance of the algorithms shown in Table 3.2. For the simulations, we use as before an SVF with cut-off frequency  $\lambda_{svf} = 3.2$  rad/s, which is slightly higher than the maximum system bandwidth. The rest of the hyperparameters has been obtained by trial and error and their values are given in Table 3.2.

Firstly, we present the results using RLSSVF-FF and RLSSVF-KF. With the former approach, good results can be expected since the parameters are varying with a similar rate. The estimates obtained using RLSSVF-FF and RLSSVF-KF are shown in Figure

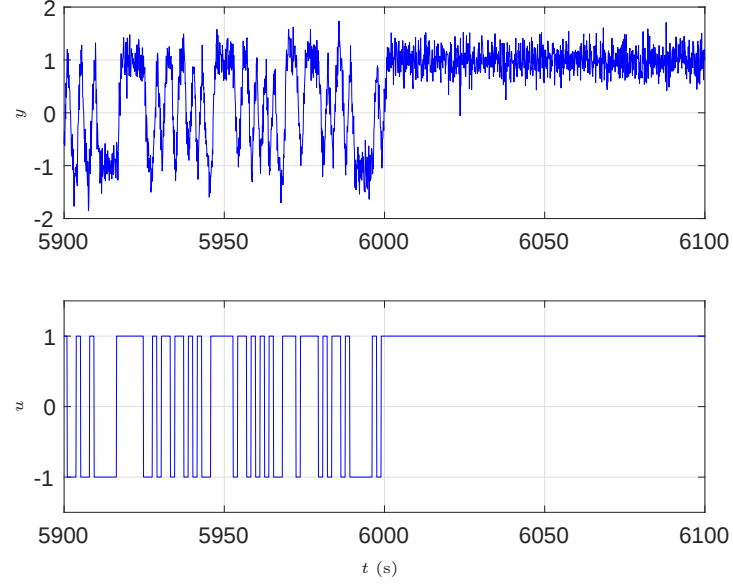


Figure 3.10: Part of the input-output data.

Table 3.2: Hyperparameters of the algorithms.

| Method     | Hyperparameters                                                                      |
|------------|--------------------------------------------------------------------------------------|
| RLSSVF-FF  | $\lambda = 0.998$                                                                    |
| RLSSVF-KF  | $Q_n = \text{diag}([10^{-4} \quad 10^{-4} \quad 10^{-4}])$                           |
| RSRIVC-KF  | $Q_n = \text{diag}([10^{-4} \quad 10^{-4} \quad 10^{-4}]); t_{s,iv} = 100 \text{ s}$ |
| LMR-RLSSVF | $\lambda = 0.998; \Gamma = \text{diag}([1 \quad 1 \quad 1])$                         |

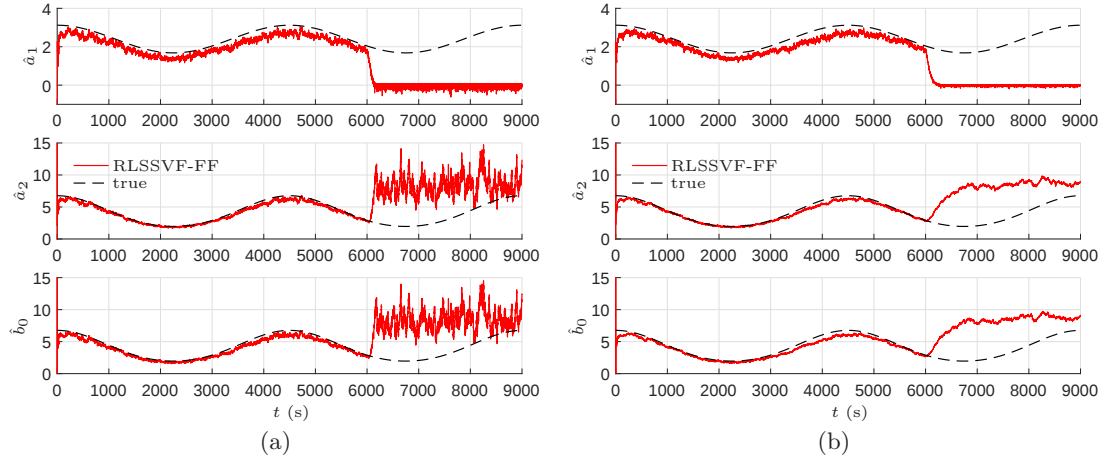


Figure 3.11: True parameters, RLSSVF-FF estimates (a) and RLSSVF-KF estimates (b) when poor excitation appear from  $t \geq 6000$  s.

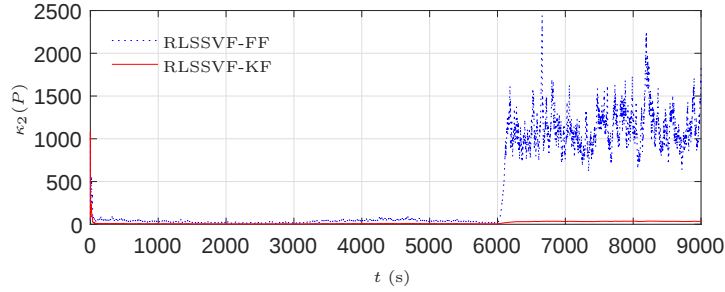


Figure 3.12: Condition number of  $P(t_k)$  for RLSSVF-FF and RLSSVF-KF when poor excitation appear from  $t \geq 6000$  s.

3.11. Under rich excitation conditions, the RLSSVF-FF estimates are similar to the ones provided by RLSSVF-KF. However this is not the case under poor excitation conditions.

The condition numbers of  $P$  are plotted in Figure 3.12. We can see a faster growth of  $\kappa_2(P)$  for RLSSVF-FF; as we already mentioned, this is expected since in forgetting factor based algorithms  $P(t_k)$  may grow exponentially under poor excitation, while in KF based algorithms the growth is linear.

The estimation with RSRIVC is performed considering a conditional updating based on  $\kappa_2(P)$  of RLSSVF-FF with a threshold  $\gamma = 300$  (see (3.79)). In Figure 3.13, the RSRIVC estimates are plotted. For  $t \approx 6000$  s, the condition  $\kappa_2(P) < 300$  does not hold anymore, and therefore  $\hat{\theta}(t_k) = 0$ .

Regarding LMR-RLSSVF, the estimates are compared with the true values in Figure 3.14. Notice that under persistent excitation conditions, the RLSSVF-FF and LMR-RLSSVF estimates are similar. However, under poor excitation,  $\kappa_2(P)$  is smaller for LMR-RLSSVF, as it can be seen in Figure 3.15. Note though that  $\kappa_2(P)$  for RLSSVF-KF is even smaller (see Figure 3.12).

### 3.7 Conclusions

For the algorithms presented in Chapter 2, several practical aspects and implementation issues, summarized in the introduction of the chapter, have been analyzed. Next we list them together the proposed solutions.

- Choice of user hyperparameters. The most important hyperparameters are:

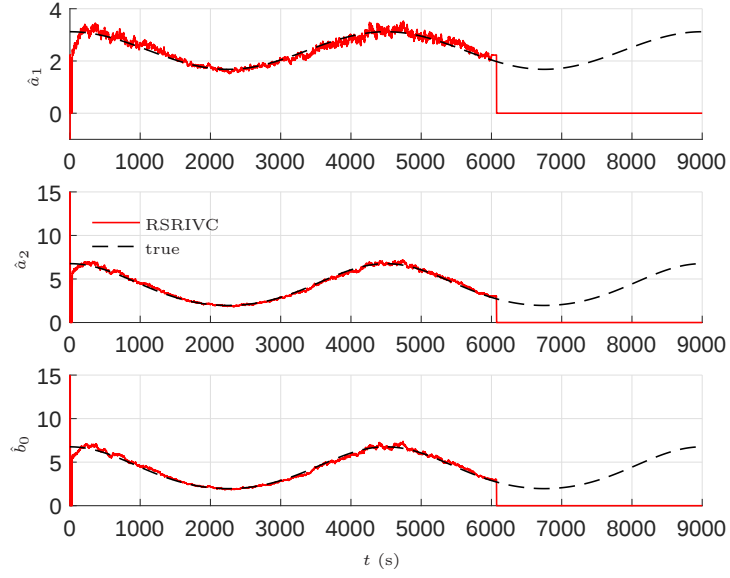


Figure 3.13: True parameters and RSRIVC estimates when poor excitation appear from  $t \geq 6000$  s.

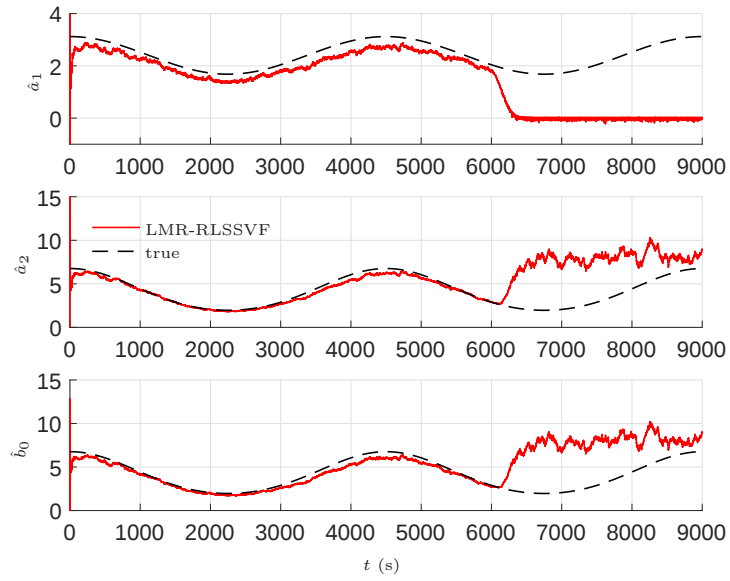


Figure 3.14: True parameters and LMR-RLSSVF estimates when poor excitation appear from  $t \geq 6000$  s

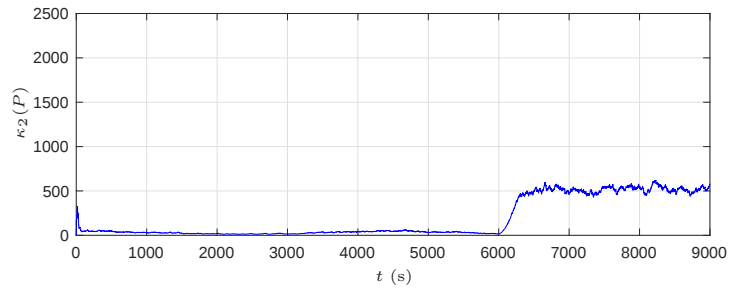


Figure 3.15: Condition number of  $P(t_k)$  for LMR-RLSSVF.

- $\lambda_{\text{svf}}$ . This is an hyperparameter of RLSSVF which is used to initialize the IV based methods. The cut-off frequency  $\lambda_{\text{svf}}$  should be chosen close to the system bandwidth. One of the assumptions that we have made is that the parameters are slowly varying. Then, to have an idea about values for  $\lambda_{\text{svf}}$ , we propose to estimate local LTI models on portions of some training data. Nonetheless, it is important to know that this parameter has also an impact in the parameter convergence, namely, the larger the value of  $\lambda_{\text{svf}}$ , the quicker the convergence. On the other hand, in the transient mode, the larger  $\lambda_{\text{svf}}$ , the larger the overshoots in the estimates can be. Through a numerical example we have shown that one way to reduce these overshoots is to start the estimation after the transients due to initial conditions have died out. Regarding the implementation of the SVF, the ordinary version could introduce an undesired signal scaling that leads to numerical issues. Therefore we suggest to use in general the normalized version.
  - $t_{\text{s,iv}}$ . The choice of the time instant at which RLSSVF converges and the IV methods starts to operate can be done automatically. To this end, some indexes have been presented. Based on RLSSVF estimations in some training data, we suggest to choose  $t_{\text{s,iv}}$  according to prediction error  $\varepsilon(t_k)$  or output error  $\varepsilon_y(t_k)$ .
  - $Q_n$ . The normalized covariance matrix of the Kalman filter can be estimated through maximum likelihood using some training data. It might be the case that it is known in advance that some parameters do not vary in time. Then, it is important that such prior information is used in the estimation through maximum likelihood in order to simplify the problem. As in the case of the choice of  $\lambda_{\text{svf}}$ , it might be also useful to estimate local LTI models on segments of training data to find out how the parameters are varying.
- Digital implementation of the CT filtering operations involved on the prefilter and auxiliary model: in order to obtain accurate filter outputs, the sampling frequency should be as high as possible.
  - Numerical aspects: for recursive LS based methods, Potter's algorithm can be used to improve conditioning. For recursive IV based methods, this is not possible because the  $P$  matrix is non-symmetric. Therefore, the proposed solution is to use signal scaling. The advantage of this approach will be shown in Section 4.1.3.
  - Estimator windup problem: the problem of having poor excitation in the data has been considered. For IV based methods, it is proposed to use conditional updating based on the condition number of the covariance matrix  $P(t_k)$ , where  $P(t_k)$  is obtained from running in parallel RLSSVF with forgetting factor.

## Chapter 4

# Applications

Here we present two real-life problems to which we apply the approaches developed in Chapters 2 and 3. The first one is an electronic bandpass filter proposed as a benchmark both for LPV and LTV model identification [Lataire et al., 2015]. The second real life based problem is an electronic throttle control (ETC) system that we find in automobiles nowadays. In the last decades, several studies have been carried out in order to improve the performance of this device (see *e.g.* [Pavković and Deur, 2011]).

### 4.1 Benchmark data from an electronic bandpass filter

In [Lataire et al., 2015], real data collected from a second order bandpass filter has been proposed as a benchmark both for LPV model and LTV model identification. It deals with an open-loop estimation problem. In Figure 4.1, a diagram of the electronic circuit is shown. A parallel connection of an  $n$ -type J-FET transistor and a 470 k $\Omega$  resistor constitute a variable resistance, which produces the parameter variations. The scheduling variable  $r(t)$  is the gate-source voltage of the transistor, and the input  $u(t)$  and output  $y(t)$  are the voltages indicated in Figure 4.1. The noise level in the measurements is very low, with an SNR of more than 60 dB. Note that some small non-linear effects are also present.

Several scenarios, consisting on different excitations and scheduling signals are provided. To assess our approaches, we consider one scenario with smooth parameter variations. Since LTV models are identified, the measured scheduling signal is not used.

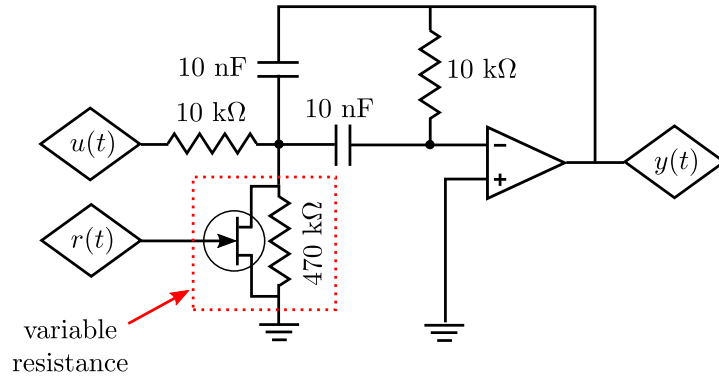


Figure 4.1: Second order bandpass filter.

#### 4.1.1 Problem formulation

According to the qualitative description of the circuit given in [Lataire et al., 2015], the system has two complex poles and one zero at the origin. Therefore, we consider the following CT LTV OE model

$$(p^2 + a_1(t)p + a_2(t))x(t) = b_0(t)pu(t) \quad (4.1a)$$

$$y(t_k) = x(t_k) + e(t_k) \quad (4.1b)$$

where the input  $u(t)$  and output  $y(t)$  are voltages. Given a data set, the aim is then to identify the parameters using the methods proposed previously in Chapters 1 and 2. One of the assumption that we make to develop those approaches is that the parameters are slowly varying. Thus, from the many different scenarios available in [Lataire et al., 2015]<sup>1</sup>, we consider the data which satisfied that assumption. The name of the chosen data set is

$$\text{MS\_Harm\_h3\_N15640\_RMS70\_P2P700.mat} \quad (4.2)$$

which means the following:

- multisine excitation
- scheduling signal composed of one harmonic
- $N = 15640$  samples (in one period)
- RMS of 70 mV for the excitation
- scheduling signal with a peak-to-peak value of 700 mV

The data consists in 6 experiments, denoted here by  $Ei$ , with  $i = 1, \dots, 6$ . Each experiment has 3 periods, with  $N = 15640$  samples in each period. In each experiment the input is different. Regarding the scheduling variable, there is one for the first 3 experiments, and another one for the last 3 experiments. We are using only the following experiments:

- $E1$  as estimation data
- $E3$  as validation data
- $E4$  as training data

Then, experiment  $E1$  of the data set (4.2) will be denoted as (4.2)- $E1$ .

The sampling frequency is 156250 Hz. Part of the input-output data is presented in Figure 4.2. The scheduling signal  $r(t)$  of the second period is shown in Figure 4.3. As we already mentioned,  $r(t)$  is not used in the identification. However it will be used to interpret the results.

To get a better insight of the system, we will also use the data set named

$$\text{MS\_Const\_N15640\_RMS70\_P2P700.mat} \quad (4.3)$$

which includes experiments with constant scheduling variables with different values.

---

<sup>1</sup>The data can be downloaded from  
[www.kth.se/social/group/system-identificatio/page/17th-ifac-symposium-on-system-identifica/](http://www.kth.se/social/group/system-identificatio/page/17th-ifac-symposium-on-system-identifica/)

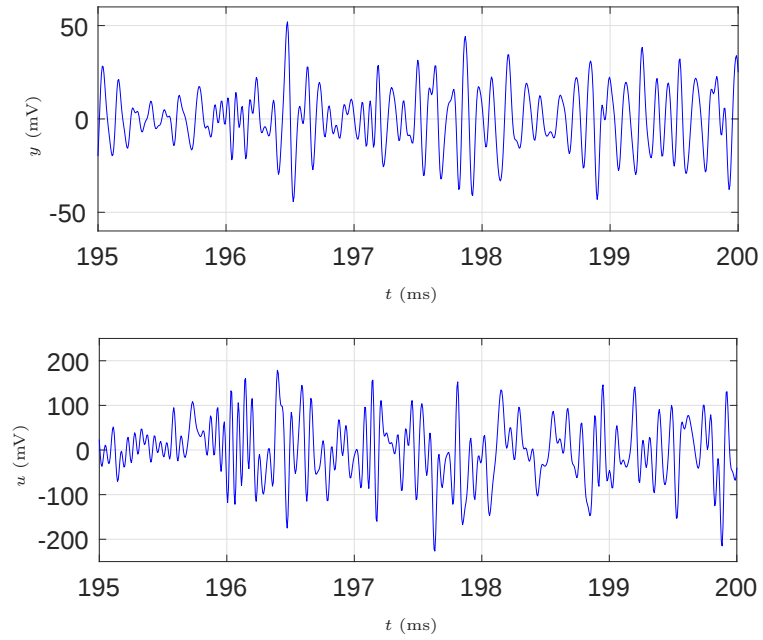


Figure 4.2: Part of the input-output data for the identification of the circuit model.

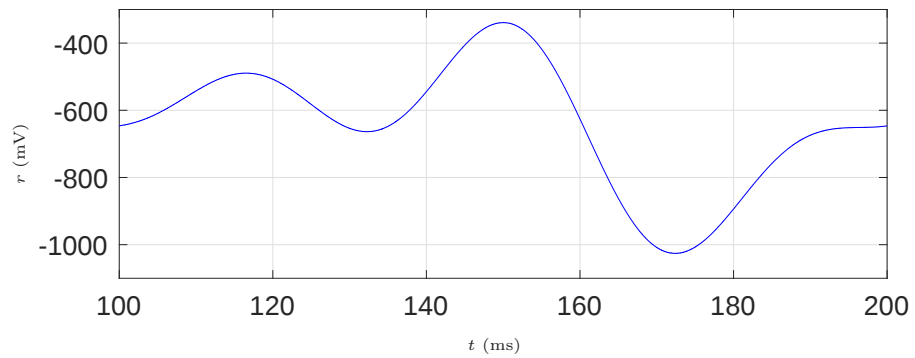


Figure 4.3: Scheduling signal  $r(t)$  of the estimation data for the identification of the circuit model.

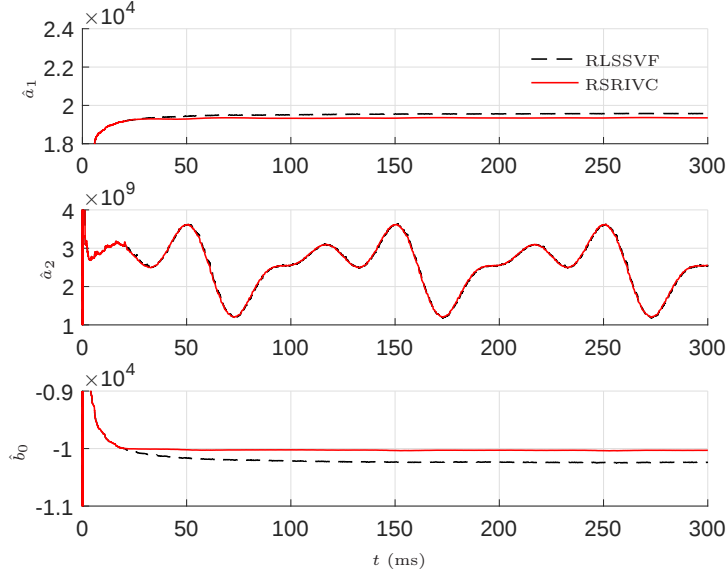


Figure 4.4: RLSSVF and RSRIVC estimates of the electrical circuit.

#### 4.1.2 Recursive estimation of the electrical circuit

Our goal is to assess the performance of RLSSVF and RSRIVC. To get rough estimates of the hyperparameters  $\lambda_{\text{svf}}$  and  $Q_n$ , we first identify LTI models using the data set (4.3) corresponding to LTI systems. The estimation is performed with SRIVC with detrended data, *i.e.* the mean is removed. This procedure lead to  $\lambda_{\text{svf}} = 10^7$  rad/s. From [Lataire et al., 2015], we know that the parameter  $b_0(t)$  is constant. Additionally we found out that also  $a_1(t)$  is nearly constant. This means that in  $Q_n$  the corresponding values for  $a_1(t)$  and  $b_0(t)$  are zero.

From the previous analysis, we know that there is only one value of  $Q_n$  remaining to be estimated. This is done using the data set (4.2)-E1 corresponding to the LTV system. As part of the preprocessing, this data is also detrended. By trial and error we finally obtain  $Q_n = \text{diag}([10^{-15} \ 0.5 \ 10^{-15}])$ . Regarding  $t_{\text{s,iv}}$ , from the convergence of RLSSVF, we choose  $t_{\text{s,iv}} = 20$  ms.

The RLSSVF estimates and RSRIVC estimates are compared in Figure 4.4. As expected, the results are very close due to the low noise level in the measurements. Notice that the variation of  $\hat{a}_2(t)$  matches the shape of the scheduling signal  $r(t)$  displayed in Figure 4.3 for one period.

The frequency response functions for the frozen RSRIVC models corresponding to the second period are shown in Figure 4.5. Note that the shape is similar to the one reported in [Lataire et al., 2015]; as it is pointed out there, the evolution of the resonance frequency (peak of the frequency response function) matches the shape of the scheduling signal (see Figure 4.3).

To assess the quality of the estimates we use the benchmark criterion proposed in [Lataire et al., 2015], which corresponds to the RMS error computed on the second period of the simulated output signal, *i.e.*

$$\text{RMSE}_y = \sqrt{\frac{1}{N - N_{\text{tr}}} \sum_{k=N+N_{\text{tr}}}^{2N-1} (y(t_k) - \hat{y}(t_k, \hat{\theta}, u))^2} \quad (4.4)$$

where  $y$  is the measured output from the validation data. The simulated output  $\hat{y}$  is computed from the estimate  $\hat{\theta}$  and the measured input from the validation data  $u$ . The term  $N_{\text{tr}} = 500$  is used to eliminate the transient effects. Additionally, we compute the

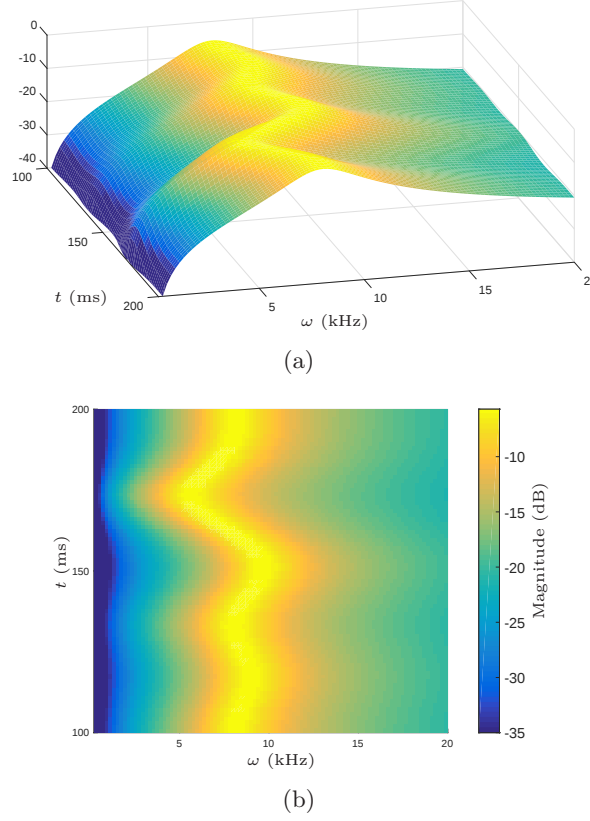


Figure 4.5: Frequency response functions of the frozen models obtained with RSRIVC.

fit between the measured and simulated outputs, which is defined as follows

$$\mathcal{F} = 100 \times \left[ 1 - \frac{\|y(t_k) - \hat{y}(t_k, \hat{\theta}, u)\|}{\|y(t_k) - \text{mean}(y(t_k))\|} \right] \quad (4.5)$$

with  $t_k$  varying as in (4.4). In [Lataire et al., 2015], for the analyzed data set, the proposed model structure, which is different than ours (see (4.1)), has 3 poles and one zero, with none of the parameters in the numerator being set to zero. Considering the data set (4.2)-E3 as validation data, we obtain after detrending the results that are shown in Table 4.1. From the values of  $\text{RMSE}_y$ , we can see that RSRIVC is doing better than RLSSVF but still not as good as the result given in [Lataire et al., 2015]. Notice that in our case, apart from having a different model structure, we are only considering the data set (4.2)-E1 as estimation data. Nevertheless, from the fits we can conclude that the estimated models are a fair representation of the system dynamics. For RSRIVC, a comparison between measured and simulated outputs is shown in Figure 4.6.

#### 4.1.3 Illustration of the signal scaling approach

The advantage of using the signal scaling approach introduced in Section 3.5 is illustrated here. Signal scaling improves the conditioning of the proposed recursive algorithms, that is, it reduces the condition number of the  $P$  matrix.

The condition numbers of  $P$  corresponding to the previous estimations using RLSSVF and RSRIVC are shown in Figures 4.7(a) and (c), respectively. The large values obtained for  $\kappa_2(P)$  can be reduced through the signal scaling approach. We should keep in mind that by using signal scaling, the parameter estimates are not altered. The signal

Table 4.1: Performance indexes using validation data.

| Method                 | $\text{RMSE}_y$ (mV) | $\mathcal{F}$ (%) |
|------------------------|----------------------|-------------------|
| [Lataire et al., 2015] | 0.1156               | not reported      |
| RLSSVF                 | 0.6909               | 95.7              |
| RSRIVC                 | 0.3781               | 97.6              |

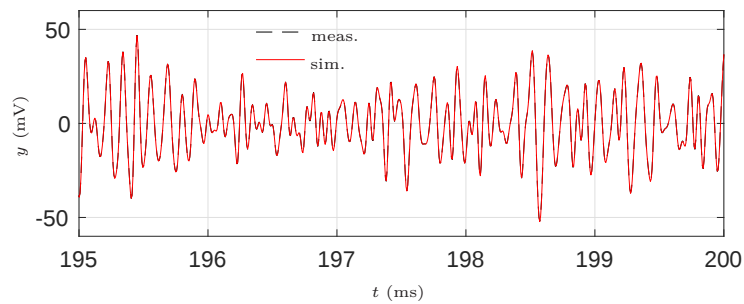


Figure 4.6: Comparison between measured and simulated outputs for RSRIVC (the measured and simulated outputs are nearly matching).

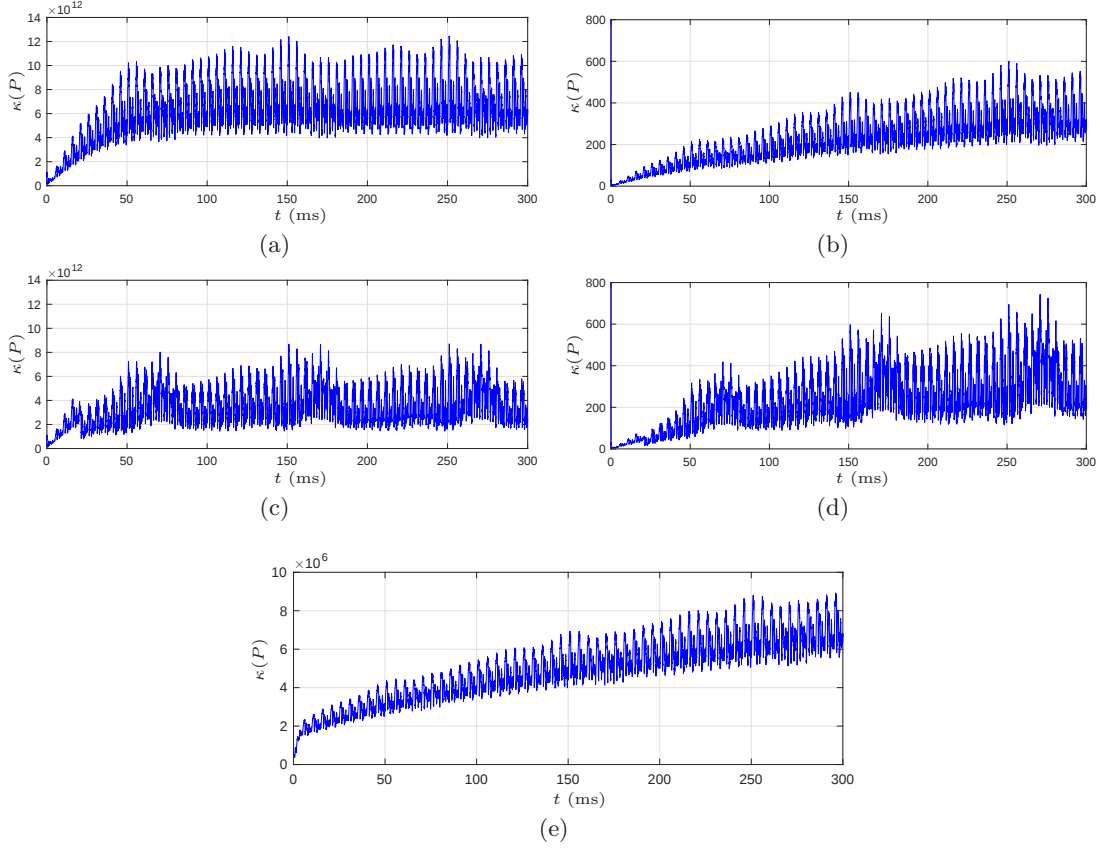


Figure 4.7: Condition number of  $P$  for (a) RLSSVF, (b) RLSSVF with signal scaling, (c) RSRIVC, (d) RSRIVC with signal scaling and (e) PotterSVF. Note that the limits for the  $y$ -axes are different.

scaling approach requires the computation of the scaling matrix  $M_s$  (see (3.53)), which is computed using the data set (4.2)- $E4$  after being detrended. The condition numbers of  $P$  with RLSSVF and RSRIVC with signal scaling decrease considerably, as we can see in Figure 4.7(b) and (d). Additionally, we provide  $\kappa_2(P)$  for the PotterSVF algorithm in Figure 4.7(e); as we can note, smaller condition numbers are obtained with the signal scaling approach.

## 4.2 Electronic throttle control system

In this section, the goal is to apply our proposed recursive estimation approaches to a simulated case study of an electronic throttle control (ETC). This corresponds to a closed-loop identification problem.

An ETC is a system used in automobiles to regulate air flow in the internal combustion engine. Basically, this is achieved by means of a DC (or step) motor which controls the position of the throttle valve plate. The ETC replaces the mechanical link between the vehicle acceleration pedal and the throttle valve in conventional automobiles. The benefits of the ETC, found both in gasoline and diesel engines, is to improve fuel economy, emissions and drivability. Additionally, the ETC is used for torque based engine control, where the desired torque is set by the pedal and the engine control system determines the necessary engine charge air and fuel to meet the torque requirement [Canudas de Wit et al., 2001, Pavković and Deur, 2011, Zhang et al., 2015].

An ETC system consists basically of a controller and an electronic throttle body

(ETB), which is shown in Figure 4.8. Apart from the DC motor and throttle plate, there are a gearbox and two springs; to simplify the scheme, only one spring is shown in Figure 4.8. While the aim of the former is to reduce the speed coming from the motor, the aim of the latter is to return the plate into the so-called limp-home (LH) position in case of power supply failure [Pavković and Deur, 2011]. Non-linearities in the system come from the springs and also from friction effects. Additionally, there is the disturbance due to the air flow. On the other hand, the physical process parameters can vary, for instance, due to [Pavković et al., 2006]:

- Aging effect.
- Variations of external temperature which may produce changes in the armature resistance of the motor.

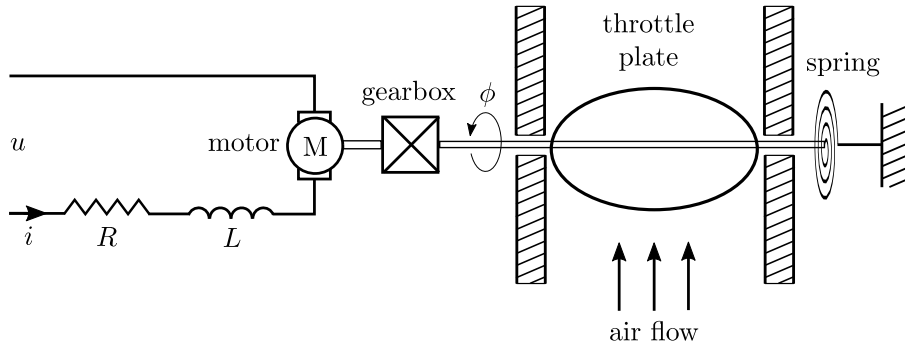


Figure 4.8: Electronic throttle body (ETB).

The design of an ETC system is a challenging task, because of the ETB's nonlinearities and parameter variations. Among different approaches, a popular method for this application is sliding control (see *e.g.* [Reichhartinger and Horn, 2009]). PID based controllers are also an option. In [Deur et al., 2004] for instance, a PID with compensation mechanisms for the nonlinearities has been developed. PIDs coupled with an adaptive mechanism have been also considered in [Canudas de Wit et al., 2001, Pavković et al., 2006].

In this chapter, we focus on the recursive estimation of the ETB parameters which could be used in an adaptive controller.

#### 4.2.1 Electronic throttle body model

The present study is done based on the model provided by [di Bernardo et al., 2010], which represents an ETB for a gasoline engine. The ETB model is given by

$$\frac{di(t)}{dt} = -\frac{R}{L}i(t) - \frac{K_v G_r}{L}\omega(t) + \frac{1}{L}u(t) \quad (4.6a)$$

$$\omega(t) = \frac{d\phi(t)}{dt} \quad (4.6b)$$

$$J\frac{d\omega(t)}{dt} = K_t G_r i(t) - M_s(t) - M_f(t) - M_a(t) \quad (4.6c)$$

Equation (4.6a) corresponds to the electrical submodel and (4.6c) to the mechanical submodel. In Table 4.2, the variables and physical parameters are described. The numerical values of the physical parameters in (4.6) are given in Table 4.3. A block diagram of the ETB model is presented in Figure 4.9, where we can distinguish the electrical and mechanical submodels.

Table 4.2: Description of ETB model parameters.

| Symbol   | Description                            |
|----------|----------------------------------------|
| $\phi$   | Angular position of the throttle plate |
| $\omega$ | Angular velocity of the throttle plate |
| $i$      | Current                                |
| $u$      | Voltage source of the armature         |
| $L$      | Inductance of the armature coil        |
| $R$      | Resistance of the armature coil        |
| $K_v$    | Velocity coefficient                   |
| $K_t$    | Torque coefficient                     |
| $J$      | Moment of inertia                      |
| $G_r$    | Transmission ratio due to the gear     |
| $M_s$    | Spring torque                          |
| $M_f$    | Friction torque                        |
| $M_a$    | Aerodynamic torque due to the air flow |

Table 4.3: ETB physical parameter values coming from [di Bernardo et al., 2010].

| Symbol              | Value                               | Unit              |
|---------------------|-------------------------------------|-------------------|
| $K_{s1}$            | $1.83 \cdot 10^{-3} \cdot 180/\pi$  | N·m/rad           |
| $K_{s2}$            | $22.90 \cdot 10^{-3} \cdot 180/\pi$ | N·m/rad           |
| $K_{s3}$            | $47.15 \cdot 10^{-3} \cdot 180/\pi$ | N·m/rad           |
| $M_{\text{close}}$  | $28.4162 \cdot 10^{-3}$             | N·m               |
| $M_{\text{open}}$   | $15.8202 \cdot 10^{-3}$             | N·m               |
| $\phi_{\text{min}}$ | $8 \cdot \pi/180$                   | rad               |
| $\phi_{\text{max}}$ | $90 \cdot \pi/180$                  | rad               |
| $\phi_{\text{LH}}$  | $12.58 \cdot \pi/180$               | rad               |
| $\Delta\phi$        | $1.12 \cdot \pi/180$                | rad               |
| $\phi_{12}$         | $85 \cdot \pi/180$                  | rad               |
| $J$                 | $1.56 \cdot 10^{-3}$                | Kg·m <sup>2</sup> |
| $K_b$               | $4.98 \cdot 10^{-3}$                | N·m·s/rad         |
| $M_c$               | $8 \cdot 10^{-3}$                   | N·m               |
| $R$                 | 1.4                                 | $\Omega$          |
| $L$                 | $2.5 \cdot 10^{-3}$                 | H                 |
| $K_v$               | 0.02                                | V·s/rad           |
| $K_t$               | $10.8856 \cdot 10^{-3}$             | N·m/A             |
| $G_r$               | 22.56                               | 1                 |

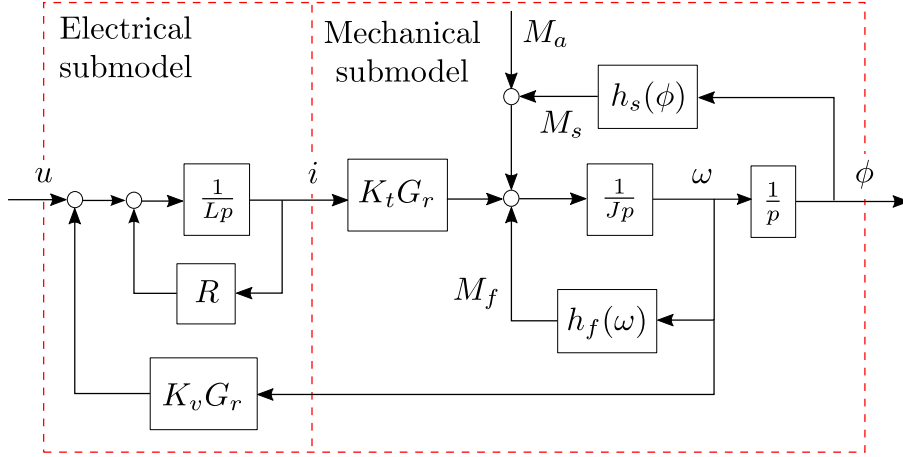
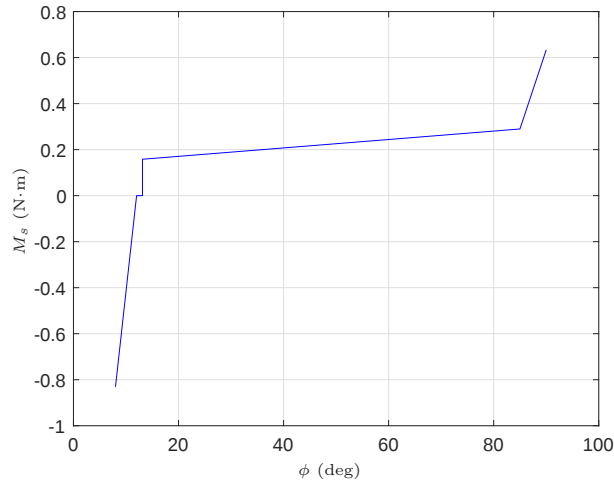


Figure 4.9: Block diagram of the ETB model.


 Figure 4.10: Spring torque  $M_s$  vs angular position  $\phi$ .

The spring torque  $M_s(t)$  is a piece-wise linear function. Here  $M_s(t)$  is considered as defined in [di Bernardo et al., 2010], *i.e.*

$$M_s(t) = h_s(\phi(t)) = \begin{cases} h_{s3}(\phi(t)) & \text{if } \phi \in [\phi_{\min}; \phi_{\text{LH}} - \Delta\phi/2[ \\ 0 & \text{if } \phi \in [\phi_{\text{LH}} - \Delta\phi/2; \phi_{\text{LH}} + \Delta\phi/2] \\ h_{s1}(\phi(t)) & \text{if } \phi \in ]\phi_{\text{LH}} + \Delta\phi/2; \phi_{12}[ \\ h_{s2}(\phi(t)) & \text{if } \phi \in [\phi_{12}; \phi_{\max}] \end{cases} \quad (4.7)$$

with

$$h_{s1}(\phi(t)) = K_{s1} \left[ \phi(t) - \left( \phi_{\text{LH}} + \frac{\Delta\phi}{2} \right) \right] + M_{\text{open}} \quad (4.8a)$$

$$h_{s2}(\phi(t)) = K_{s2}(\phi(t) - \phi_{12}) + h_{s1}(\phi_{12}) \quad (4.8b)$$

$$h_{s3}(\phi(t)) = -K_{s3} \left[ \phi_{\text{LH}} - \frac{\Delta\phi}{2} - \phi(t) \right] - M_{\text{close}} \quad (4.8c)$$

The values of the coefficients in (4.7) and (4.8) are given in Table 4.3. A plot of  $M_s(t)$  is presented in Figure 4.10.

The literature about friction modeling is extensive (see *e.g.* [Olsson et al., 1998] and references therein). To represent friction in an ETB, both static and dynamical models have been used. In [di Bernardo et al., 2010], the static model consists of Coulomb, Stribeck and viscous friction. Regarding a dynamic model, the so-called LuGre model is considered in [Canudas de Wit et al., 2001]. In our case, the model proposed in [di Bernardo et al., 2010] is simplified, such that the Stribeck effect is neglected. Then,  $M_f(t)$  is given by

$$\begin{aligned} M_f(t) &= h_f(\omega(t)) \\ &= K_b\omega(t) + M_c\text{sign}(\omega(t)) \end{aligned} \quad (4.9)$$

with the coefficients defined in Table 4.3.

The aerodynamic torque  $M_a(t)$  depends *e.g.* on the throttle plate angle  $\phi(t)$  among others [Canudas de Wit et al., 2001]. Here we consider that it is zero.

#### 4.2.2 Problem formulation

We define now the true system for the simulations based on the model (4.6). As it was mentioned before, the system is subject to parameter variations. Some changes can be very slow due to aging effects. However, this is not the case for the armature resistance, which can differ considerably within a single cycle of engine operation, due to temperature variations in the DC windings [Pavković and Deur, 2011, p. 97]. According to [Pavković and Deur, 2011, p. 98], the armature resistance does not exceed  $\pm 50\%$  of the nominal value. The same variation is assumed here, *i.e.*

$$0.5 \cdot R \leq R^\circ(t) \leq 1.5 \cdot R \quad (4.10)$$

with the nominal value  $R$  specified in Table 4.3. The rest of the true parameters, denoted also with a superscript  $\{\}^\circ$ , are defined according to Table 4.3.

Since the electrical dynamics is much faster than the mechanical dynamics, the dynamics of the current is usually neglected. Such an assumption is also considered here both for the true system and the model to be presented next. In this way we can better evaluate the tracking of the parameters. Then, the true system  $\mathcal{S}$  is given by

$$i(t) = \frac{1}{R^\circ(t)}u(t) - \frac{K_v^\circ G_r^\circ}{R^\circ(t)}\omega(t) \quad (4.11a)$$

$$\omega(t) = \frac{d\phi(t)}{dt} \quad (4.11b)$$

$$J^\circ \frac{d\omega(t)}{dt} = K_t^\circ G_r^\circ i(t) - M_s(t) - M_f(t) \quad (4.11c)$$

$$\phi_m(t_k) = \phi(t_k) + e_o(t_k) \quad (4.11d)$$

with  $R^\circ(t)$  varying linearly in the range (4.10). Regarding the signal  $e_o(t_k)$ , it is a zero-mean DT Gaussian noise sequence with variance  $10^{-3} \text{ (rad/s)}^2$ . The torques  $M_s(t)$  and  $M_f(t)$  are also defined considering the values provided in Table 4.3 as the true parameters.

The system operates in closed loop, and the closed-loop configuration is shown in Figure 4.11. The true plant  $\mathcal{G}_o$  is defined by the first three equations in (4.11).  $F_\phi$  is a filter to smooth the set-point and  $\mathcal{C}$  is a PID controller. The former is described by

$$F_\phi(q^{-1}) = \frac{0.009901 + 0.009901q^{-1}}{1 - 0.9802q^{-1}} \quad (4.12)$$

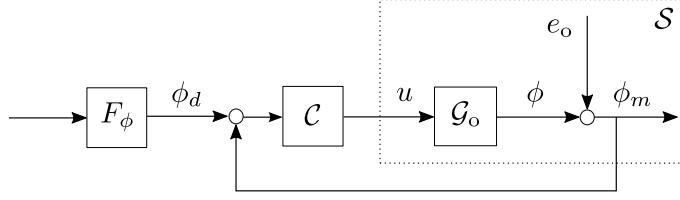


Figure 4.11: Closed-loop block diagram of the ETB.

and the latter by

$$\mathcal{C} : C_c(q^{-1}) = k_p + \frac{k_i}{1 - q^{-1}} + k_d(1 - q^{-1}) \quad (4.13)$$

with  $k_p = 2.85$ ,  $k_i = 6.06 \cdot 10^{-3}$ ,  $k_d = -36.96$ .

Due to the complexity of the system, the model identification of a throttle valve is done considering more than one experiment (see *e.g.* [Isermann and Münchhof, 2010, Pavković and Deur, 2011]). In this study, assuming that the parameters of the mechanical submodel have been previously identified, we focus on the recursive estimation of the electrical submodel which is given by

$$i(t) = \frac{1}{R(t)}u(t) - \frac{K_v G_r^o}{R(t)}\omega(t) \quad (4.14a)$$

$$\omega(t) = \frac{d\phi(t)}{dt} \quad (4.14b)$$

$$\phi_m(t_k) = \phi(t_k) + e(t_k) \quad (4.14c)$$

with  $e(t_k)$  a zero-mean DT Gaussian noise sequence. Note that both the electrical and mechanical subsystem have the parameter  $G_r^o$ , and it is assumed to be known. The parameter vector to be estimated is

$$\begin{aligned} \theta(t) &= [\theta_1(t) \quad \theta_2(t)]^T \\ &= \left[ \frac{1}{R(t)} \quad \frac{K_v G_r^o}{R(t)} \right]^T \end{aligned} \quad (4.15)$$

To avoid non-linearities coming from the spring, the estimation data satisfies the condition

$$\phi \in ]\phi_{LH} + \Delta\phi/2; \phi_{12}[ = ]13.14; 85[ \quad (\text{deg}) \quad (4.16)$$

*i.e.* we need to identify the linear function  $h_{s1}(\phi(t))$  defined in (4.8a). Therefore, the non-linear effects are only due to friction. Notice that the closed-loop system is excited only through the set point  $\phi_d(t)$ , *i.e.* no external signal<sup>2</sup> is considered, since then it is easier to satisfy (4.16) by a proper choice of  $\phi_d(t)$ .

Then, the identification problem is to recursively estimate the time-varying parameter vector (4.15) that characterizes the model (4.14), based on the sequences

$$\{\phi_d(t_k), u(t_k), i(t_k), \phi_m(t_k)\}_{k=1}^{N'}$$

where  $N'$  is the number of samples which increases by one with every recursion.

### 4.2.3 Recursive estimation of the electrical submodel

Let us formulate the solution with CLRSRIVC. Firstly, note that (4.14) is not a CT LTV OE model, which is the type of model considered in the development of the identification

<sup>2</sup>The external signal corresponds to  $r_1(t)$  in Figure 2.15.

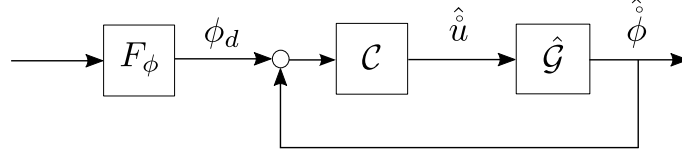


Figure 4.12: Closed-loop auxiliary model.

methods in Chapter 2. However, the proposed methods can be adapted to handle this situation.

From a heuristic approach we found out that a suitable prefilter for the identification of the electrical model is the prefilter corresponding to the mechanical system, which can be written as follows

$$\left(p^2 + \frac{K_b^o}{J^o}p + \frac{K_{s1}^o}{J^o}\right)\phi(t) = \frac{K_t^o G_r^o}{J^o}i(t) - \frac{m_{s1}^o}{J^o} - \frac{M_c^o}{J^o}\text{sign}(\omega(t)) \quad (4.17)$$

with  $p$  the differentiation operator, and  $m_{s1}^o$  gathering the constant terms in (4.8a), *i.e.* it is given by

$$m_{s1}^o = -K_{s1}^o \left( \phi_{\text{LH}}^o + \frac{\Delta\phi^o}{2} \right) + M_{\text{open}}^o \quad (4.18)$$

Note that if we discard the last term in (4.17), which is the source of the non-linearity, we obtain a multi-input single-output linear time-varying (MISO LTV) model with 2 inputs. The prefilter corresponding to this MISO LTV model is

$$F(p) = \frac{K_{s1}^o/J^o}{p^2 + (K_b^o/J^o)p + K_{s1}^o/J^o} \quad (4.19)$$

Since we assume that the parameters of the mechanical subsystem are known and constant, then the prefilter (4.19) is also known and constant.

Now we rewrite (4.14) as a filtered linear regression. By assuming slow parameter variations, we can neglect the non-commutativity issue and then apply the prefilter (4.19) to (4.14a) obtaining

$$i_f(t) = \theta_1(t)u_f(t) - \theta_2(t)\phi_f^{(1)}(t) \quad (4.20)$$

The closed-loop auxiliary model to compute the instruments is shown in Figure 4.12, where the estimated plant  $\hat{\mathcal{G}}$  is given by

$$\hat{i}(t) = \frac{1}{\hat{R}(t)}\hat{u}(t) - \frac{\hat{K}_v G_r^o}{\hat{R}(t)}\hat{\omega}(t) \quad (4.21a)$$

$$\hat{\omega}(t) = \frac{d\hat{\phi}(t)}{dt} \quad (4.21b)$$

$$J^o \frac{d\hat{\omega}(t)}{dt} = K_t^o G_r^o \hat{i}(t) - \hat{M}_s(t) - \hat{M}_f(t) \quad (4.21c)$$

with the torques  $\hat{M}_s(t)$  and  $\hat{M}_f(t)$  defined with the true parameters. The other chosen hyperparameters are:  $t_{s,\text{iv}} = 5$  s;  $Q_n = \text{diag}([10^{-6} \quad 5 \cdot 10^{-5}])$ .

For comparative purposes we also consider RLSSVF with the SVF

$$F(p) = \left( \frac{\lambda_{\text{svf}}}{p + \lambda_{\text{svf}}} \right)^2 \quad (4.22)$$

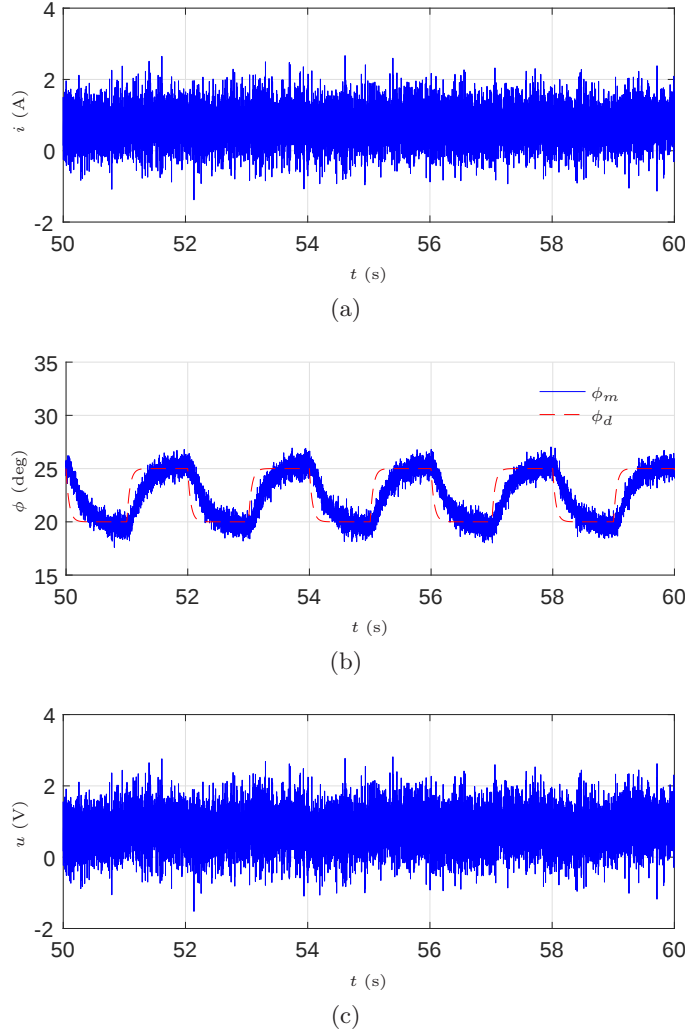


Figure 4.13: Estimation data for the identification of the ETB model.

where  $\lambda_{\text{svf}}$  is defined in terms of the bandwidth  $\omega_b$  of (4.17), which is 12.4 (rad/s); the chosen value is  $\lambda_{\text{svf}} = 5.0 \cdot \omega_b$ . For  $Q_n$  we take the same matrix as for CLRSRIVC.

We present next some results for a single experiment run out of the 100 that are considered later in a Monte Carlo simulation analysis. The measurements that are used for the model estimation are shown in Figure 4.13. The results for RLSSVF and CLRSRIVC are presented in Figures 4.14 (a) and (b), respectively. We can clearly see that while for RLSSVF there is a significant tracking error, for CLRSRIVC the estimates are very close to the true values. From  $\hat{\theta}(t)$  it is straightforward to compute the physical estimates  $\hat{R}(t)$  and  $\hat{K}_v$  (see (4.15)). A comparison between the true physical parameters and the CLRSRIVC estimates is shown in Figure 4.15.

The Monte Carlo simulation is performed with 100 experiments. The results, presented in Figure 4.16, confirm the previous analysis for a single experiment.

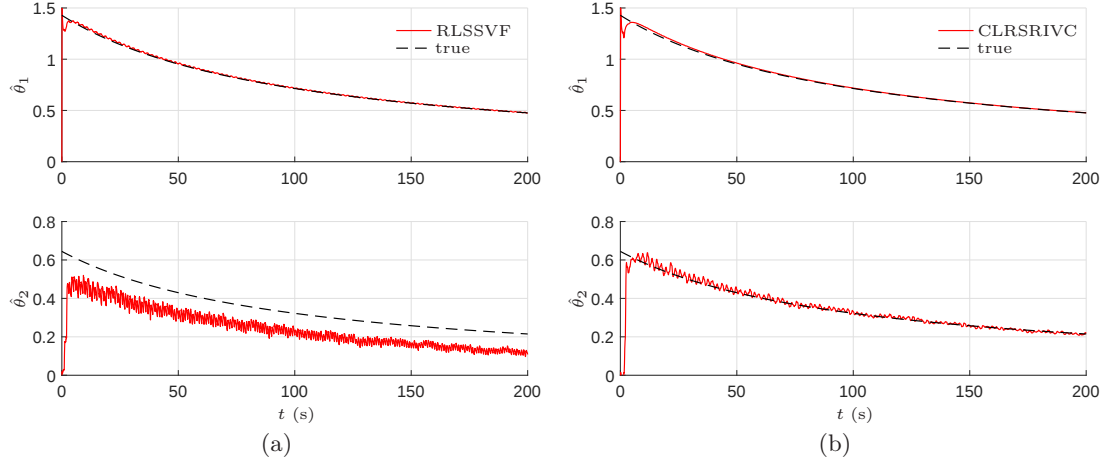


Figure 4.14: True parameters, RLSSVF and CLRSRIVC estimates for a single experiment.

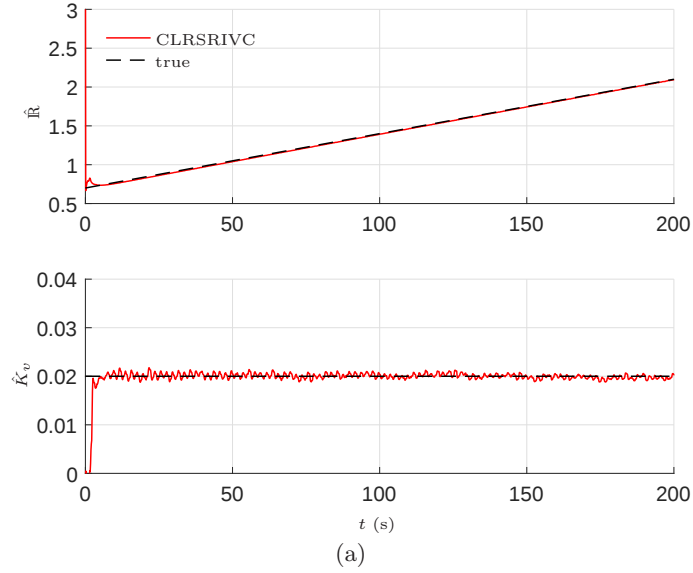


Figure 4.15: True physical parameters and CLRSRIVC estimates for a single experiment.

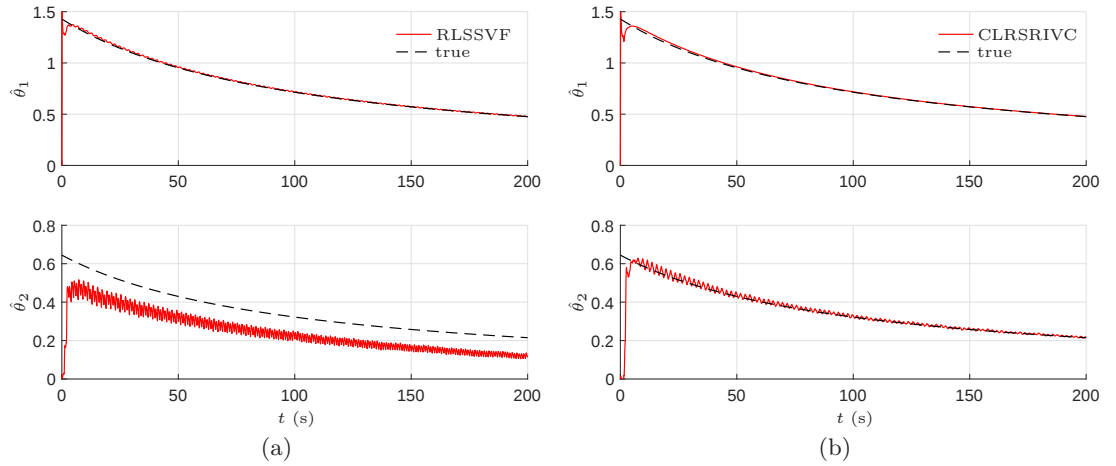


Figure 4.16: True parameters, RLSSVF and CLRSRIVC average estimates for a MC simulation.

## Chapter 5

# Conclusions and future work

### 5.1 Conclusions

In this thesis we have addressed the on-line identification of CT LTV models. In particular we focused on CT output-error type models whose parameters vary slowly in time.

In Chapter 1, we have first shown how linear filter methods are able to estimate CT LTI models. A simple approach that has been recalled is the LSSVF method which gives a closed form solution. That is what makes this least squares based method attractive. However, if the output measurement noise is not negligible, the LSSVF estimates will be biased. For the identification of CT LTV models, the recursive counterpart of LSSVF, denoted RLSSVF, can be used. In LTV model identification, analogously to the LTI case, the presence of measurement noise deteriorates the RLSSVF estimates, and the degradation of the results might not be acceptable for certain applications.

In Chapter 2, the contribution has been to develop recursive instrumental variable based methods, both for open-loop and closed-loop identification of CT LTV models. As RLSSVF, these are also direct identification methods based on the linear filter approach, but provide better estimates when the output measurement noise is not negligible. To track the time-varying parameter variations, we proposed the forgetting factor and the random walk model approaches. Nevertheless, we have focused on the latter since it is more flexible, meaning that it is suitable for cases in which the parameters vary at different rates. Furthermore, it should be more robust in cases of poor excitation. Numerical examples including Monte Carlo simulations have been considered to show that IV methods are advantageous over RLSSVF in terms of delivering parameter estimates with smaller tracking errors. One of the numerical examples and some of the algorithms developed in this chapter, have been included in the CONTSID toolbox.

In Chapter 3 we have proposed solutions to the practical aspects and implementation issues related to the use of the recursive IV methods that have been developed. The main contributions are:

- Development of a method to estimate the normalized covariance matrix of the Kalman filter  $Q_n$ . This is one of the hyperparameters of the developed identification methods that has to be chosen by the user. The tracking ability and noise sensitivity of Kalman filter based algorithms depend on  $Q_n$ . If the parameters of the system are not varying at the same speed, different values for the diagonal elements of  $Q_n$  have to be chosen. Finding suitable values for the covariance matrix might be time-consuming and not straightforward. We have proposed an off-line method to estimate  $Q_n$  from a given training data set. The approach couples the maximum likelihood method with one of the developed IV based methods. The effectiveness of this approach has been shown in a numerical example.

- Proposition of guidelines to choose the hyperparameters of the developed identification methods. Guidelines were given for the following hyperparameters, which along with  $Q_n$ , are the most important:
  - Cut-off frequency of the SVF ( $\lambda_{svf}$ ). This is an hyperparameter of RLSSVF which is used to initialize the IV based methods. The impact of  $\lambda_{svf}$  in the estimates have been discussed. In particular, we showed through a numerical example how overshoots of the estimates in the transient mode are related to  $\lambda_{svf}$ . Approaches to reduce these overshoots have been also presented.
  - Time instant when the IV based method starts to operate ( $t_{s,iv}$ ). The value of  $t_{s,iv}$  should be the time instant at which the RLSSVF has converged. Several measures that can be used to choose  $t_{s,iv}$  were discussed.
- Proposition of an anti-windup approach for IV based methods. When a system to be identified is poorly excited, anti-windup methods are needed to prevent the parameter estimates to blow up. For the IV based methods, the solution that we have proposed is to use conditional updating based on the condition number of the parameter covariance matrix  $P$ , where  $P$  is obtained from the RLSSVF with forgetting factor algorithm running in parallel. The motivation to use the RLSSVF with forgetting factor method is that it should be in general more sensitive than other algorithms to poor excitation.
- Development of a signal scaling approach to improve conditioning of IV based methods. Numerical aspects can play an important role when recursive algorithms are run in a microcontroller. To improve the conditioning of the identification problem and thus to avoid numerical issues, we have proposed to use a signal scaling approach. The effectiveness of this method has been demonstrated with a benchmark considering real data from an electronic band pass filter.

Finally, in Chapter 4, a simulation case study corresponding to the estimation of an electronic throttle valve model is considered. The identification problem is challenging because of non-linear effects due to friction. Moreover the system operates in closed-loop. We have illustrated how to adapt one of the developed IV methods to the recursive estimation of the physical time-varying parameters.

## 5.2 Future work

In this thesis we have developed recursive instrumental variable methods for CT models under the assumption that the parameters are slowly varying in time. This assumption allows us to neglect the non-commutativity problem when linear filter method are applied. Additionally this assumption is used to guarantee stability of the auxiliary model. The goal of a future work can be to adapt the developed approaches to scenarios with fast or abrupt parameter variations. The problem is not trivial because of these non-commutativity and stability issues.

For fast parameter variations, a solution might be to use a parallel estimation scheme, where several recursive algorithms running in parallel compete with each other. These algorithms could differ in the assumed representation of the time-varying parameters which can be described by different deterministic and/or stochastic models. For the case of abrupt parameter variations, parallel estimation is also an option.

One of our algorithms, developed under the assumption of a having a CT LTV system, has been adapted to a CT non-linear time-varying system. Another possible direction of research would be to develop algorithms specially dedicated to this type of systems.

## Appendix A

# Digital implementation of CT filtering operations

The digital implementation issues of the CT filtering operations are well-known in CT model identification. They should be treated in an appropriate way since errors generated by the digital implementation can have an impact on the quality of the estimated model.

The CT filtering operation is represented by a LTV state-space system

$$\dot{x}(t) = F(t)x(t) + G(t)u(t) \quad (\text{A.1a})$$

$$y(t) = H(t)x(t) + J(t)u(t) \quad (\text{A.1b})$$

with state  $x(t) \in \mathbb{R}^{n_x}$ , input  $u(t) \in \mathbb{R}^{n_u}$ , output  $y(t) \in \mathbb{R}^{n_y}$  and matrices  $F(t)$ ,  $G(t)$ ,  $H(t)$  and  $J(t)$  of appropriate dimensions. Depending on the intersample behaviour of the input  $u(t)$ , two different discretizations are considered in this work.

### A.1 Zero-order-hold assumption

The zero-order-hold assumption is considered if the intersample behaviour of the input  $u(t)$  is piecewise constant. In the LTV scenario, the zero-order-hold assumption applies also to the system time-varying matrices  $F(t)$ ,  $G(t)$ ,  $H(t)$  and  $J(t)$ , i.e. the following is satisfied for the input

$$u(t) = u(t_k), \quad \forall t \in [(k-1)T_s, kT_s) \quad (\text{A.2})$$

and for the time-varying matrices we have

$$F(t) = F(t_k) \quad (\text{A.3a})$$

$$G(t) = G(t_k) \quad (\text{A.3b})$$

$$H(t) = H(t_k) \quad (\text{A.3c})$$

$$J(t) = J(t_k), \quad \forall t \in [(k-1)T_s, kT_s) \quad (\text{A.3d})$$

That is, between samples the matrices are constant. As pointed out in [Tóth et al., 2010] in the context of LPV systems, this assumption of having constant matrices is usually not realistic. The discrete-time version of (A.1) is then defined as in the LTI case (see [Åström and Wittenmark, 1996, p. 34]) and it is given by

$$x(t_{k+1}) = F_d(t_k)x(t_k) + G_d(t_k)u(t_k) \quad (\text{A.4a})$$

$$y(t_k) = H_d(t_k)x(t_k) + J_d(t_k)u(t_k) \quad (\text{A.4b})$$

where

$$F_d(t_k) = e^{T_s F((k-1)T_s)} \quad (\text{A.5a})$$

$$G_d(t_k) = \int_0^{T_s} e^{F((k-1)T_s)s} ds G \quad (\text{A.5b})$$

$$H_d(t_k) = H((k-1)T_s) \quad (\text{A.5c})$$

$$J_d(t_k) = J((k-1)T_s) \quad (\text{A.5d})$$

These matrices  $F_d(t_k)$  and  $G_d(t_k)$  can be computed from

$$\begin{bmatrix} F_d(t_k) & G_d(t_k) \\ 0 & I \end{bmatrix} = \exp \left( \begin{bmatrix} F((k-1)T_s) & G((k-1)T_s) \\ 0 & 0 \end{bmatrix} T_s \right) \quad (\text{A.6})$$

with ‘exp’ corresponding to the matrix exponential that can be computed in MATLAB with `expm`.

## A.2 Trapezoidal approximation

If the intersample behaviour for the input  $u(t)$  is different than piecewise constant, the trapezoidal approximation may be used assuming that the system time-varying matrices  $F(t)$ ,  $G(t)$ ,  $H(t)$  and  $J(t)$  remain constant between samples, i.e. (A.3) is satisfied.

The discrete-time version of (A.1) is then defined as in the LTI case (see [Franklin et al., 1998, p. 200]) and it given by (A.4), where

$$F_d(t_k) = \left( I + \frac{T_s}{2} F((k-1)T_s) \right) \left( I - \frac{T_s}{2} F((k-1)T_s) \right)^{-1} \quad (\text{A.7a})$$

$$G_d(t_k) = \sqrt{T_s} \left( I - \frac{T_s}{2} F((k-1)T_s) \right)^{-1} G((k-1)T_s) \quad (\text{A.7b})$$

$$H_d(t_k) = \sqrt{T_s} H((k-1)T_s) \left( I - \frac{T_s}{2} F((k-1)T_s) \right)^{-1} \quad (\text{A.7c})$$

$$J_d(t_k) = J((k-1)T_s) + \frac{T_s}{2} H((k-1)T_s) \left( I - \frac{T_s}{2} F((k-1)T_s) \right)^{-1} G((k-1)T_s) \quad (\text{A.7d})$$

## Appendix B

# Stability of LTV systems

The following material is extracted mainly from [Khalil, 2002, pp. 156-158]. Let us consider the LTV system

$$\dot{x}(t) = F(t)x(t) \quad (\text{B.1})$$

with state  $x(t) \in \mathbb{R}^{n_x}$  and system matrix  $F(t)$ . The solution of (B.1) is given by

$$x(t) = \Phi(t, t_0)x(t_0)$$

where  $\Phi(t, t_0)$  is the state transition matrix. The following theorem defines stability in terms of the state transition matrix.

**Theorem 1** *The equilibrium point  $x = 0$  of (B.1) is uniformly exponentially stable if and only if the state transition matrix satisfies the inequality*

$$\|\Phi(t, t_0)\| \leq ke^{\lambda(t-t_0)}, \quad \forall t \geq t_0 \geq 0 \quad (\text{B.2})$$

for some positive constants  $k$  and  $\lambda$ .

To test stability in practice, instead of using Theorem 1, it is more convenient to work with the following theorem.

**Theorem 2** *Let,*

- $x = 0$  be the exponentially stable equilibrium point of (B.1).
- $Q(t)$  be a continuous, bounded, positive definite, symmetric matrix.

*Then, there is a continuously differentiable, bounded, positive definite, symmetric matrix  $P(t)$  that satisfies*

$$-\dot{P}(t) = P(t)F(t) + F^T(t)P(t) + Q(t) \quad (\text{B.3})$$

*Hence,  $V(t, x) = x^T P(t)x$  is a Lyapunov function for the system that satisfies the conditions of Theorem 1.*

Equation (B.3) corresponds to a linear matrix inequality that can be solved numerically.

## Appendix C

# Scaled version of Potter's algorithm

For simplicity of exposition, the time index is omitted. To distinguish  $\{\cdot\}(t_k)$  from  $\{\cdot\}(t_{k-1})$ , for the former we use the superindex  $\{\cdot\}^+$ . Then, the a-posteriori covariance matrix is written as follows

$$P^+ = P - \frac{P\varphi_f\varphi_f^T P}{1 + \varphi_f^T P \varphi_f} \quad (\text{C.1})$$

Replacing (3.65b) in (C.1) we obtain

$$M_s P_s^+ M_s = M_s P_s M_s - \frac{M_s P_s M_s \varphi_f \varphi_f^T M_s P_s M_s}{1 + \varphi_f^T M_s P_s M_s \varphi_f} \quad (\text{C.2})$$

Using the square root factorization  $P_s = S_s S_s^T$  yields

$$M_s S_s^+ S_s^{+T} M_s = M_s S_s S_s^T M_s - \frac{M_s S_s S_s^T M_s \varphi_f \varphi_f^T M_s S_s S_s^T M_s}{1 + \varphi_f^T M_s S_s S_s^T M_s \varphi_f} \quad (\text{C.3})$$

If we consider

$$U = M_s S_s \quad (\text{C.4a})$$

$$g = S_s^T M_s \varphi_f = U^T \varphi_f \quad (\text{C.4b})$$

then (C.3) can be written in the form of (3.69), *i.e.*

$$\begin{aligned} U^+ U^{+T} &= U U^T - \frac{U g g^T U^T}{\beta} \\ &= U \left[ I_d - \frac{g g^T}{\beta} \right] U^T \end{aligned} \quad (\text{C.5})$$

Since (C.5) has the same form than (C.3), we know that

$$\beta = 1 + g^T g \quad (\text{C.6})$$

, and  $U$  can be written as follows

$$U^+ = U \left[ I_d - \frac{g g^T}{\alpha} \right] \quad (\text{C.7})$$

where  $\alpha$  is given by (3.72) but with  $\beta$  defined in (C.6). Finally, considering (C.4a), the square root of  $P_s$  can be recursively computed as follows

$$S_s^+ = S_s \left[ I_d - \frac{g g^T}{\alpha} \right] \quad (\text{C.8})$$

---

Then, the scaled version of the Kalman filter based Potter's algorithm is given by

*Prediction step:*

$$\hat{\theta}_s(t_k|t_{k-1}) = \hat{\theta}_s(t_{k-1}) \quad (\text{C.9a})$$

$$\begin{bmatrix} S_s^T(t_k|t_{k-1}) \\ 0 \end{bmatrix} = T \begin{bmatrix} S_s^T(t_{k-1}) \\ Q_{n,s}^{T/2} \end{bmatrix} \quad (\text{C.9b})$$

*Correction step:*

$$\hat{\theta}_s(t_k) = \hat{\theta}_s(t_k|t_{k-1}) + M_s^{-1}L(t_k)\varepsilon(t_k) \quad (\text{C.9c})$$

$$\varepsilon(t_k) = y_f^{(na)}(t_k) - \varphi_f^T(t_k)M_s\hat{\theta}_s(t_k|t_{k-1}) \quad (\text{C.9d})$$

$$g(t_k) = S_s^T(t_k|t_{k-1})M_s\varphi_f(t_k) \quad (\text{C.9e})$$

$$\beta(t_k) = 1 + g^T(t_k)g(t_k) \quad (\text{C.9f})$$

$$\alpha(t_k) = \beta(t_k) + \sqrt{\beta(t_k)} \quad (\text{C.9g})$$

$$L(t_k) = \frac{M_s S(t_k|t_{k-1}) S^T(t_k|t_{k-1}) M_s \varphi_f(t_k)}{\beta(t_k)} \quad (\text{C.9h})$$

$$S_s(t_k) = S_s(t_k|t_{k-1}) \left[ I_d - \frac{g(t_k)g^T(t_k)}{\alpha(t_k)} \right] \quad (\text{C.9i})$$

Notice that (C.9) involves many multiplications with the diagonal matrix  $M_s$ , which means that several multiplications with zeros are performed. A more efficient scalar implementation could be developed.

# Bibliography

- K. Åstrom. Maximum likelihood and prediction error methods. *Automatica*, 1980.
- K. Åstrom and B. Wittenmark. *Computer-Controlled Systems*. Prentice Hall, 3rd edition, 1996.
- K. Åstrom and B. Wittenmark. *Adaptive Control*. Dover Publications Inc., 2008.
- V. Bavdekar, A. Deshpande, and S. Patwardhan. Identification of process and measurement noise covariance for state and parameter estimation using extended Kalman filter. *Journal of Process Control*, 21(4):585–601, 2011.
- M. Bergamasco and M. Lovera. Continuous-time predictor-based subspace identification using Laguerre filters. *IET Control Theory and Applications*, 5(7):856–867, 2011.
- A. Bittencourt, A. Isakson, D. Peretzki, and K. Forsman. An algorithm for finding process identification intervals from normal operating data. *Processes - Open Access Journal*, 2015.
- C. Canudas de Wit. Recursive estimation of the continuous-time process parameters. In *25th Conference on Decision and Control*, pages 2016–2020, Athens, Greece, 1986.
- Carlos Canudas de Wit, I. Kolmanovsky, and Sun J. Adaptive pulse control of electronic throttle. In *American Control Conference*, pages 2872–2877, Arlington, US, 2001.
- L. Cao and H. Schwartz. A novel recursive algorithm for directional forgetting. In *American Control Conference*, San Diego, California, US, 1999.
- L. Cao and H. Schwartz. A directional forgetting algorithm based on the decomposition of the information matrix. *Automatica*, 36:1725–1731, 2000.
- L. Cao and H. Schwartz. Analysis of the Kalman filter based estimation algorithm: an orthogonal decomposition approach. *Automatica*, 40:5–19, 2004.
- S. Cao and R.R. Rhinehart. An efficient method for on-line identification of steady state. *Journal of Process Control*, 5(6):363–374, 1995.
- S.C. Chan and Z.G. Zhang. Local polynomial modeling and variable bandwidth selection for time-varying linear systems. *IEEE Transactions on Instrumentation and Measurement*, 60(3):1102–1117, 2011.
- T.B. Co and S. Ungarala. Batch scheme recursive parameter estimation of continuous-time systems using the modulating function method. *Automatica*, 33(6):1185–1191, 1997.
- S. Dasgupta and Y.F. Huang. Asymptotically convergent modified recursive least-squares with data-dependent updating and forgetting factor for systems with bounded noise. *IEEE Transactions on Information Theory*, IT-33(3):383–392, 1987.

- J. Deur, D. Pavković, N. Perić, M. Jansz, and D. Hrovat. An electronic throttle control strategy including compensation of friction and limp-home effects. *IEEE Transactions on Industry Applications*, 40(3):821–834, 2004.
- M. di Bernardo, A. di Gaeta, U. Montanaro, and S. Santini. Synthesis and experimental validation of the novel LQ-NEMCSI adaptive strategy on an electronic throttle valve. *IEEE Transactions on Control Systems Technology*, 18(6):1325–1337, 2010.
- D. Dimogianopoulos and R. Lozano. Adaptive control for linear slowly time-varying systems using direct least-squares estimation. *Automatica*, 37:251–256, 2001.
- M. Evestedt. *Parameter and State Estimation with Information-rich Signals*. PhD thesis, Uppsala University, 2007.
- M. Evestedt, A. Medvedev, and T. Wigren. Windup properties of recursive parameter estimation algorithms in acoustic echo cancellation. *Control Engineering Practice*, 16: 1372–1378, 2008.
- T.I. Fossen. *Marine control systems*. Trondheim: Marine Cybernetics, 2002.
- G. Franklin, J. Powell, and M. Workman. *Digital Control of Dynamic Systems*. Addison-Wesley, 3rd edition, 1998.
- B. Fridholm, T. Wik, and M. Nilsson. Robust recursive impedance estimation for automotive lithium-ion batteries. *Journal of Power Sources*, 304:33–41, 2016.
- H. Garnier. Direct continuous-time approaches to system identification. Overview and benefits for practical applications. *European Journal of Control*, 24:50–62, 2015.
- H. Garnier and P. C. Young. The advantages of directly identifying continuous-time transfer function models in practical applications. *International Journal of Control*, 87(7):1319–1338, 2014.
- H. Garnier, P. Sibilie, and T. Spott. Influence of the initial covariance matrix on recursive LS estimation of continuous models via generalised Poisson moment functionals. In *IFAC Symposium on System Identification*, Copenhagen, Denmark, 1994.
- H. Garnier, M. Mensler, and A. Richard. Continuous-time Model identification from sampled data: Implementation issues and performance evaluation. *International Journal of Control*, 76(13):1337–1357, 2003.
- H. Garnier, G. C. Goodwin, and J. Welsh. A time-domain approach to continuous-time model identification of highly resonant wide-band systems. Technical report, Centre de Recherche en Automatique de Nancy, 2004.
- H. Garnier, L. Wang, and P. C. Young. Direct identification of continuous-time models from sampled data: issues, basic solutions and relevance. In *Identification of continuous-time models from sampled data*, volume H. Garnier and L. Wang (Eds.), pages 1–30. Springer, 2008.
- M. Gilson and P. Van den Hof. Instrumental variable methods for closed-loop system identification. *Automatica*, 41:241–249, 2005.
- M. Gilson, H. Garnier, P. C. Young, and P. Van den Hof. Instrumental variable methods for closed-loop continuous-time model identification. In *Identification of Continuous-time Models from Sampled Data*, volume H. Garnier and L. Wang (Eds.), pages 133–160. Springer, 2008.

- M. Gilson, H. Garnier, P. C. Young, and P.M.J. Van den Hof. Optimal instrumental variable methods for closed-loop identification. *IET Control Theory and Applications*, 5(10):1147–1154, 2011.
- G. C. Goodwin and K. S. Sin. *Adaptive Filtering Prediction and Control*. Dover Publications Inc., 1984.
- M. S. Grewal and A. P. Andrews. *Kalman Filtering. Theory and Practice using MATLAB*. Wiley, 2015.
- S. Gunnarsson. Combining tracking and regularization in recursive least squares identification. In *Proceedings of the 35th IEEE Conference on Decision and Control*, pages 2551–2552, 1996.
- F. Gustafsson. *Adaptive Filtering and Change Detection*. John Wiley & Sons, 2000.
- L. Gustafsson and M. Olsson. Robust on-line estimation. Master’s thesis, Lund Institute of Technology, 1999.
- S. Haykin. *Adaptive Filter Theory*. Pearson, 5th edition, 2014.
- A. Ilchmann, D.H. Owens, and D. Prätzel-Wolters. Sufficient conditions for stability of linear time-varying systems. *System & Control Letters*, 9:157–163, 1987.
- R. Isermann. Fault diagnosis of machines via parameter estimation and knowledge processing - Tutorial paper. *Automatica*, 29(4):815–835, 1993.
- R. Isermann and M. Münchhof. *Identification of Dynamic Systems. An Introduction with Applications*. Springer-Verlag, 2010.
- Z.H. Jiang and W. Schaufelberger. A recursive identification method for continuous time-varying linear systems. In *34th Midwest Symposium on Circuits and Systems*, volume 1, pages 436–439, Monterey, CA, USA, 1991.
- R. Johansson. Identification of continuous-time models. *IEEE Transactions on Signal Processing*, 42(4):887–897, 1994.
- H.K. Khalil. *Nonlinear Systems*. Prentice Hall, 2002.
- G. Kitagawa and W. Gersch. A smoothness priors-state space modeling of time series with trend and seasonality. *Journal of the American Statistical Association*, 79(386): 378–389, 1984.
- I. Landau, M. Alma, A. Constantinescu, J. Martinez, and M. Noë. Adaptive regulation - Rejection of unknown multiple narrow band disturbances (a review on algorithms and applications). *Control Engineering Practice*, 19:1168–1181, 2011.
- J. Lataire. *Frequency Domain Measurement and Identification of Linear, Time-Varying Systems*. PhD thesis, Vrije Universiteit Brussel, 2011.
- J. Lataire, E. Louarroudi, R. Pintelon, and Y. Rolain. Benchmark data on a linear time- and parameter-varying system. In *IFAC Symposium on System Identification*, Beijing, China, 2015.
- V. Laurain, R. Tóth, M. Gilson, and H. Garnier. Direct identification of continuous-time linear parameter-varying input/output models. *IET Control Theory and Applications*, 5(7):878–888, 2011.

- L. Lindbom. *A Wiener Filtering Approach to the Design of Tracking Algorithms With Applications in Mobile Radio Communications*. PhD thesis, Uppsala University, 1995.
- L. Ljung. *System Identification. Theory for the User*. Prentice Hall, Upper Saddle River, 2nd edition, 1999.
- L. Ljung and S. Gunnarsson. Adaptation and tracking in system identification - a survey. *Automatica*, 26(1):7–21, 1990.
- L. Ljung and T. Söderström. *Theory and Practice of Recursive Identification*. The MIT Press, Cambridge, MA, 1983.
- M. Niedźwiecki. *Identification of Time-varying Processes*. Wiley, 2000.
- M. Niedźwiecki and T. Klaput. Fast recursive basis function estimators for time-varying processes. *IEEE Transactions on Signal Processing*, 50(8):1925–1934, 2002.
- H. Olsson, K. Åström, C. Canudas de Wit, M. Gäfvert, and P. Linschinsky. Friction models and friction compensation. *European Journal of Control*, 4:176–195, 1998.
- M. Östring and S. Gunnarsson. Recursive identification of physical parameters in a flexible robot arm. *Asian Journal of Control*, 6(3):407–414, 2004.
- A. Padilla, H. Garnier, and M. Gilson. Version 7.0 of the CONTSID toolbox. In *Symposium on System Identification*, Beijing, China, 2015.
- A. Padilla, H. Garnier, P. C. Young, and J. Yuz. Real-time identification of continuous-time linear time-varying systems. In *IEEE 55th Conference on Decision and Control*, pages 3769–3774, Las Vegas, US, 2016.
- A. Padilla, H. Garnier, P. C. Young, and J. I. Yuz. Recursive online IV method for identification of continuous-time slowly time-varying models in closed loop. In *IFAC World Congress*, Toulouse, France, 2017.
- J. E. Parkum. *Recursive Identification of time-varying systems*. PhD thesis, Technical University of Denmark, 1992.
- J. E. Parkum, N. K. Poulsen, and J. Holst. Recursive forgetting algorithms. *International Journal of Control*, 55(1):109–128, 1992.
- D. Pavković and J. Deur. *Modeling and Control of Electronic Throttle Drive*. Lambert Academic Publishing, 2011.
- D. Pavković, J. Deur, M. Jansz, and N. Perić. Adaptive control of automotive electronic throttle. *Control Engineering Practice*, 14:121–136, 2006.
- D. Peretzki, A.J. Isaksson, A.C. Bittencourt, and K. Forsman. Data mining of historic data for process identification. In *Proc. of the 2011 AIChE Annual Meeting*, 2011.
- G. Pillonetto, F. Dinuzzo, T. Chen, G. De Nicolao, and L. Ljung. Kernel methods in system identification, machine learning and function estimation: a survey. *Automatica*, 50:657–682, 2014.
- G. Prando, D. Romeres, and A. Chiuso. Online identification of time-varying systems: a Bayesian approach. In *IEEE 55th Conference on Decision and Control*, pages 3775–3780, Las Vegas, US, 2016.

- F. Previdi and M. Lovera. Identification on non-linear parametrically varying models using separable least squares. *International Journal of Control*, 77:1382–1392, 2004.
- A. Quarteroni, R. Sacco, and F. Saleri. *Numerical Mathematics*. Springer, 2000.
- M. Reichhartinger and M. Horn. Application of higher order sliding-mode concepts to a throttle actuator for gasoline engines. *IEEE Transactions on Industrial Electronics*, 56(9):3322–3329, 2009.
- C. Rojas. *Robust Experiment Design*. PhD thesis, University of Newcastle, 2008.
- W. J. Rugh. *Linear System Theory*. Prentice Hall, 1996.
- J. Rutström. *Simplified Wiener LMS Tracking with Automatic Tuning of the Step-Size*. PhD thesis, Uppsala University, 2005.
- S. Saelid and B. Foss. Adaptive controller with a vector variable forgetting factor. In *22nd IEEE Conference on Decision and Control*, pages 1488–1494, San Antonio TX, US, 1983.
- M. Salgado, G. Goodwin, and R. Middleton. Modified least squares algorithm incorporating exponential resetting and forgetting. *International Journal of Control*, 47(2): 477–491, 1988.
- A. H. Sayed and T. Kailath. A state-space approach to adaptive RLS filtering. *IEEE Signal Processing Magazine*, 11(3):18–60, July 1994. (Expanded version).
- D. Sbarbaro, P. Ascencio, P. Espinoza, F. Mujica, and G. Cortes. Adaptive soft-sensors for on-line particle size estimation in wet grinding circuits. *Control Engineering Practice*, 16:171–178, 2008.
- Y.A.W. Shardt and B. Huang. Data quality assessment of routine operating data for process identification. *Computers and Chemical Engineering*, 2013.
- D. Simon. *Optimal State Estimation. Kalman,  $H_\infty$ , and Nonlinear Approaches*. John Wiley & Sons, 2006.
- J. Sjöberg, T. McKelvey, and L. Ljung. On the use of regularization in system identification. In *IFAC World Congress*, pages 318–386, Sydney, Australia, 1993.
- T. Söderström and P. Stoica. *Instrumental Variable Methods for System Identification*. Springer-Verlag, 1983.
- T. Söderström and P. Stoica. *System Identification*. Prentice Hall, 1989.
- G. Solbrand, A. Ahlén, and L. Ljung. Recursive methods for off-line identification. *International Journal of Control*, 41(1):177–191, 1985.
- B. Stenlund and F. Gustafsson. Avoiding windup in recursive parameter estimation. In *Preprints of reglermöte*, Sweden: Linköping, 2002.
- P. Stoica and P. Åhgren. Exact initialization of the recursive least squares. *International Journal of Adaptive Control and Signal Processing*, 16:219–230, 2002.
- R. Tóth. *Modeling and Identification of Linear Parameter-Varying Systems*. Springer, 2010.

- R. Tóth, P.S.C. Heuberger, and P.M.J. Van den Hof. Discretisation of linear parameter-varying state-space representations. *IET Control Theory and Applications*, 4(10): 2082–2096, 2010.
- R. Tóth, V. Laurain, M. Gilson, and H. Garnier. Instrumental variable scheme for closed-loop LPV model identification. *Automatica*, 48:2314–2320, 2012.
- S. Ungarala and T.B. Co. Time-varying system identification using modulating function and spline models with application to bio-processes. *Computer & Chemical Engineering*, 24(12):2739–2753, 2000.
- P. Van den Hof and B. Ninness. System identification with generalized orthonormal basis functions. In P.S.C. Heuberger, P.M.J. Van de Hof, and B. Wahlberg, editors, *Modeling and Identification with Rational Orthogonal Basis Functions*, pages 61–102. Springer, 2005.
- T. Van Waterschoot, G. Rombouts, and M. Moonen. Optimal regularized adaptive filtering algorithms for room acoustic signal enhancement. *Signal Processing*, 88:594–611, 2008.
- K.Y. Wong and E. Polak. Identification of linear discrete time systems using the instrumental variable method. *IEEE Transactions on Automatic Control*, AC-12(6): 707–718, 1967.
- Ch. K. Yoo, S. W. Sung, and In-B. Lee. Generalized damped least squares algorithm. *Computers & Chemical Engineering*, 27:423–431, 2003.
- P. C. Young. *Recursive Estimation and Time-Series Analysis: An Introduction*. Springer-Verlag, Berlin, 1984.
- P. C. Young. *Recursive Estimation and Time-series Analysis. An Introduction for the Student and Practitioner*. Springer-Verlag, Berlin, 2nd edition, 2011.
- P. C. Young. Refined instrumental variable estimation: Maximum likelihood optimization of a unified Box-Jenkins model. *Automatica*, 52:35–46, 2015.
- P. C. Young, D. J. Pedregal, and W. Tych. Dynamic harmonic regression. *Jnl. of Forecasting*, 18:369–394, 1999.
- P. C. Young, P. McKenna, and J. Bruun. Identification of nonlinear stochastic systems by state dependent parameter estimation. *International Journal of Control*, 74:1837–1857, 2001.
- P. C. Young, H. Garnier, and M. Gilson. Refined instrumental variable identification of continuous-time hybrid Box-Jenkins models. In *Identification of continuous-time models from sampled data*, volume H. Garnier and L. Wang (Eds.), pages 1–30. Springer, 2008.
- S. Zhang, J. Yang, and G. Zhu. LPV Modeling and mixed constrained  $H_2/H_\infty$  control of an electronic throttle. *IEEE/ASME Transactions on Mechatronics*, 20(5):2120–2132, 2015.
- Y. Zheng and Z. Lin. Recursive adaptive algorithms for fast and rapidly time-varying systems. *IEEE Transactions on Circuits and Systems*, 50(9):602–614, 2003.

## Résumé

Les travaux présentés dans ce mémoire traitent de l'identification des systèmes dynamiques représentés sous la forme de modèles linéaires continus à paramètres variant lentement au cours du temps. La complexité du problème d'identification provient d'une part du caractère inconnu de la loi de variation des paramètres et d'autre part de la présence de bruits de nature inconnue sur les signaux mesurés.

Les solutions proposées s'appuient sur une combinaison judicieuse du filtre de Kalman en supposant que les variations des paramètres peuvent être représentées sous la forme d'une marche aléatoire et de la méthode de la variable instrumentale qui présente l'avantage d'être robuste vis à vis de la nature des bruits de mesure.

Les algorithmes de type récursif sont développés dans un contexte d'identification en boucle ouverte et en boucle fermée. Les différentes variantes se distinguent par la manière dont est construit la variable instrumentale. Inspirée de la solution développée pour les systèmes linéaires à temps invariant, une construction adaptative de la variable instrumentale est suggérée pour pouvoir suivre au mieux l'évolution des paramètres. Les performances des méthodes développées sont évaluées à l'aide de simulations de Monte Carlo et montrent la suprématie des solutions proposées s'appuyant sur la variable instrumentale par rapport celles plus classiques des moindres carrés récurrents.

Les aspects pratiques et d'implantation numérique sont d'une importance capitale pour obtenir de bonnes performances lorsque ces estimateurs sont embarqués. Ces aspects sont étudiés en détails et plusieurs solutions sont proposées non seulement pour robustifier les estimateurs vis à vis du choix des hyper-paramètres mais également vis à vis de leur implantation numérique. Les algorithmes développés sont venus enrichir les fonctions de la boîte à outils CONTSID pour Matlab.

Enfin, les estimateurs développés sont exploités pour faire le suivi de paramètres de deux systèmes physiques : un benchmark disponible dans la littérature constitué d'un filtre électronique passe-bande et une vanne papillon équipant les moteurs de voiture. Les deux applications montrent le potentiel des approches proposées pour faire le suivi de paramètres physiques variant lentement dans le temps.

**Mots-clés :** modèle à temps continu, modèle linéaire à paramètres variables, estimateur de la variable instrumentale, filtre de Kalman

# Abstract

The work presented in this thesis deals with the identification of dynamic systems represented through continuous-time linear models with slowly time-varying parameters. The complexity of the identification problem comes on the one hand from the unknown character of the parameter variations and on the other hand from the presence of noises of unknown nature on the measured signals.

The proposed solutions rely on a judicious combination of the Kalman filter assuming that the variations of the parameters can be represented in the form of a random walk, and the method of the instrumental variable which has the advantage of being robust with respect to the nature of the measurement noises.

The recursive algorithms are developed in an open-loop and closed-loop identification setting. The different variants are distinguished by the way in which the instrumental variable is built. Inspired by the solution developed for time-invariant linear systems, an adaptive construction of the instrumental variable is suggested in order to be able to follow the evolution of the parameters as well as possible. The performance of the developed methods are evaluated using Monte Carlo simulations and show the supremacy of the proposed solutions based on the instrumental variable compared with the more classical least squares based approaches.

The practical aspects and implementation issues are of paramount importance to obtain a good performance when these estimators are used. These aspects are studied in detail and several solutions are proposed not only to robustify the estimators with respect to the choice of hyperparameters but also with respect to their numerical implementation. The algorithms developed have enhanced the functions of the CONTSID toolbox for Matlab.

Finally, the developed estimators are considered in order to track parameters of two physical systems: a benchmark available in the literature consisting of a bandpass electronic filter and a throttle valve equipping the car engines. Both applications show the potential of the proposed approaches to track physical parameters that vary slowly over time.

**Keywords:** continuous-time model, linear time-varying model, instrumental variable, Kalman filter