# Nonlinear system identification with kernels

Yusuf Bhujwalla

UNIVERSITÉ DE LORRAINE

École doctorale IAEM Lorraine

CRAN

# Nonlinear System Identification with Kernels

*Applications of Derivatives in Reproducing Kernel Hilbert Spaces*

# THÈSE

soutenue publiquement le 5 decembre 2017

pour l'obtention du

**Doctorat de l'Université de Lorraine**

**(mention automatique, traitement du signal et génie informatique)**

par

Yusuf Bhujwalla

**Composition du jury**

| | | |
|---|---|---|
| *Rapporteurs :* | Guillaume Mercère | Maître de Conférences HdR à l'Université de Poitiers, France |
| | Johan Suykens | Professeur à Katholieke Universiteit Leuven, Belgique |
| *Examinateurs :* | Martine Olivi | Chargé de Recherche HdR INRIA Sophia Antipolis, France |
| | Thierry Bastogne | Professeur à l'Université de Lorraine, France |
| *Encadrants :* | Vincent Laurain | Maître de Conférences à l'Université de Lorraine, France |
| | Marion Gilson | Professeur à l'Université de Lorraine, France |

**Centre de Recherche en Automatique de Nancy CRAN — UMR 7039**

Mis en page avec la classe thesul.

*To my grandparents: Granny Kathleen, Grandad Mickey, Grandad YB and Granny Fiza.*

# Remerciements

I've been very lucky to meet a great many wonderful people not only over the past three years, but also throughout my life. And I am indebted to an enourmous amount of those people - teachers, colleagues, family, friends, friends of family and family of friends. So, as much as I would like to, I can't thank everyone. That being said, I would just like to mention of one or two people in particular.

First of all, I'd like to thank Marion and Vincent.

Not only did they give me the opportunity to come here but also, they've supported me completely throughout my three years here. They've pushed me and encouraged me when required - which, it should be said, has probably been more often than they would've liked to have had to do. They've given me the opportunity to go to numerous conferences and workshops, both here in France and abroad. And in doing all of this, they've invested a lot, both professionally and personally, and both in me and this project. This has, I'm sure, required copious quantities of patience, belief, enthusiasm and faith - and some more patience besides. So, really, honestly, thank you both. And I hope that you are both as proud of this work as I am.

I would also like to thank the members of my jury: Professor Thierry Bastogne, Chargé de Recherche HdR Martine Olivi, Maître de Conférences HdR Guillaume Mercère and Professor Johan Suykens. I would like to thank you all for the time and effort you gave to review my manuscript and participate in my defence. The process of defending was something I was very nervous about - as I'm sure everyone is - but in fact it ended up being a very enjoyable experience, in large part due to the feedback and support you all provided.

I would also like to thank my family - in particular my parents (Kay and Mus) and my sister (Shireen).

My parents have always supported me, encouraged me, pushed me and - to the best of my abilities - done what they can to help me. My Mum has always been there to talk to, to read over letters of application and to provide advice as and when required. Advice without which I would have done even more stupid things than I already have. And my Dad has given up hours, days and possibly weeks of his life to help me take my books and bikes from one place to another. By boat, car or train ; in the snow and the rain ; on Christmas Eve and New Year's Day ; from and to Ireland to and from England, and from and to England to and from France.

And thanks also to my sister Shireen: thanks for going with me to the ferry three and a half years ago. And thanks for always being there: I can't imagine my life without you. Well, I say can't imagine, but I can. I'm guessing it would've been a lot more peaceful - but who would want that ?

And last but by no means least, I would also like to thank Christelle. I met Christelle in my first year here in Nancy and since then I have been lucky to have the opportunity to get to know an extraordinary and beautiful person. Thank you very much for putting up with me over the past couple of years, and in particular over the past few months.

# Sommaire

# Abstract

Continued research into methods for the identification of nonlinear systems has led to the availability of a wide repertoire of techniques [Ljung, 1999, Nelles, 2001, Tóth, 2010]. However, the open nature of the problem naturally means that there is no one optimal approach, but rather a corpus of methods each suited to particular problems [Judistky et al., 1995, Sjöberg et al., 1995].

Kernel-based approaches to nonlinear modelling have garnered increasing attention from the identification community in recent years [Pillonetto et al., 2014]. To date, most research on the subject in control and identification has focused on one of either two distinct paradigms, coming from the statistics and machine learning communities respectively : *Gaussian Processes* and *Least-Squares Support Vector Machines*.

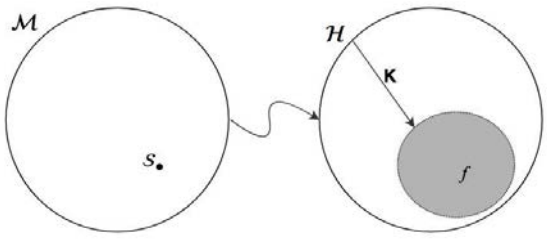



FIGURE 1 – *Identification in the RKHS : the choice of* $\mathbf{K}$ *determines the properties of the estimated model.*

A *Gaussian Process* interpretation [Rasmussen and Williams, 2006] relates the kernel function to the covariance function of a stochastic random process, in turn permitting a probabilistic interpretation of the problem.

By contrast, the *Least-Squares Support Vector Machines* approach [Suykens et al., 2002] provides a deterministic framework for estimation, considering the kernel as a mapping between a primal (input) space - in which the learning problem is formulated - and a dual (feature) space, in which the problem is solved.

Both provide a powerful framework for esitmating nonlinear models, and have been successfully applied to many problems of particular interest in system identification, e.g. [Goethals et al., 2005a, Tóth et al., 2011, Laurain et al., 2015,Darwish et al., 2015].

Recently however, attention has also begun to turn towards other kernel-based approaches, such as *Reproducing Kernel Hilbert Spaces (RKHS)* [Aronszajn, 1950]. *RKHS* methods provide a conceptual link between popular kernel approaches and classical system identification, by directly relating fundamental notions in nonlinear identification (such as the model class, model structure and complexity) to the specification of a high-dimensional nonlinear feature space $\mathcal{H}$ through the definition of a kernel function $\mathbf{K}$ (e.g. Figure 1).

Models are formed by taking linear combinations of weighted kernels, located throughout the input space.

$$\mathcal{M} : f(x) = \sum_i \alpha_i k_{x_i}(x),\ \alpha_i \in \mathbb{R}. \qquad (1)$$

In this way, by defining different kernel functions (of which many are readily avai-

lable in the literature [Poggio and Girosi, 1990, Rasmussen and Williams, 2006]) the same set of techniques can be straightforwardly transposed to a range of different problems. Such problems are inherently *ill-posed*, and require additional constraints on the behaviour of the function (i.e. regularisation terms) to ensure the uniqueness of the of solution, e.g.

$$\mathcal{J}(f) = \sum_{k=1}^{N} (y_k - f(x_k))^2 + \lambda \|f\|_{\mathcal{H}}^2, \qquad (2)$$

where $\lambda$ is used to control the bias-variance trade-off of the model.

Another increasingly popular approach to kernel-based identification builds on the *RKHS* methodology by considering the formulation of a model in a *Sobolev space* [Adams, 1975]. Sobolev space methods (e.g. smoothing splines) have emerged in recent years as another mathematically elegant and computationally efficient framework for nonlinear identification problems [Wahba, 1990, Pillonetto et al., 2014].

They allow the direct incorporation of structural constraints (particularly smoothness) into the optimisation criterion by encoding functional derivatives into the norm of the feature space, e.g.

$$\|f\|_{\mathcal{S}_k^p} = \sum_{i=0}^{k} \int_{\mathcal{X}} \left( \frac{d^i f(x)}{dx^i} \right)^p dx. \qquad (3)$$

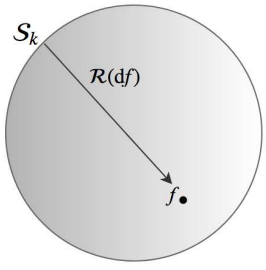



FIGURE 2 – *In Sobolev spaces, a flexible kernel function is defined with* ***soft*** *structural choices made through a regularisation term.*

For $p = 2$, $\mathcal{S}_k^2$ is a Hilbert space additionally defined through its functional derivatives. Formulating the problem in such a manner has many implications, notably :

- Though the definition of a Sobolev space, the definition of $\mathbf{K}$ is implied. Therefore, the choice of kernel is not free, but rather the optimal kernel for each problem must be determined. This results in an *a priori* flexible, hyperparameter-free kernel function.
- In this framework, in (2), $\lambda$ now not only controls the variance of the model - but also the extent to which the properties defined in (2.16) are respected in the model. Hence, *soft* constraints on the properties of the model can be continuously controlled through a regularisation hyperparameter - as opposed to *hard* constraints placed by the choice of kernel function (as illustrated in Figure 2).

However, they lack the adaptability of conventional RKHS approaches. As the kernel function is determined by solving the corresponding Green's function for the probem, determing new solutions is non-trivial in practice - in some cases even infeasible. Hence, the user is inherently limited : both in the way in which this method can be readily applied to different nonlinear problems and in what types of structural properties can be penalised.

The major contribution of this thesis is the development of a derivative regularisation approach consistent with the traditional RKHS setting : allowing *hard* structural choices through the kernel optimisation and *soft* structural choices through a regularisation term (as in Figure 3).

By directly formulating and penalising derivatives in the chosen feature space [Zhou, 2008, Bhujwalla et al., 2016a, Bhujwalla et al., 2016b,Bhujwalla et al., 2017b, Lauer et al., 2012] :

$$\mathcal{J}(f) = \sum_{k=1}^{N} (y_k - f(x_k))^2 + \lambda \left\| \frac{d^m f(x)}{dx^m} \right\|_{\mathcal{H}}^2, \qquad (4)$$

an approach is developed that can be easily implemented in practical nonlinear identification scenarios. Such an approach can allow for : the approximation of nonlinear structures, structural detection, complexity tuning, variance reduction and simplification of the hyperparameter optimisation problem.

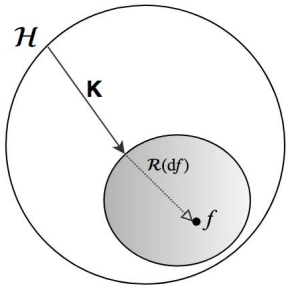



FIGURE 3 – *Formulating derivative penalities in an RKHS directly allows* ***hard*** *and* ***soft*** *optimisation of the model class.*

This method will be studied extensively in the thesis :

- A general framework for derivative regularisation in nonlinear identification will be developed.
- Theoretical and practical concerns will be addressed.
- A comparison between the developed approaches and methods from the literature will be presented.
- And the method will be applied to several problems of particular interest in nonlinear identification.

In addition to simulation examples, an application of the proposed methods to an industrial case-study will also be presented. In collaboration with *Post Luxembourg*, a tier-2 internet service provider and content delivery network, a framework was developed for the modelling and forecasting of traffic coming from and going to the network from different autonomous systems [Bhujwalla et al., 2017a].

*Keywords : nonlinear system identification, kernel methods, regularisation, reproducing kernel Hilbert spaces, derivatives in the RKHS.*

# Résumé

## Resumé de la Thèse

La poursuite des recherches sur les méthodes d'identification des systèmes non linéaires a permis de disposer d'un large répertoire de techniques [Ljung, 1999,Nelles, 2001,Tóth, 2010]. Cependant, la nature ouverte du problème signifie naturellement qu'il n'y a pas une seule approche optimale, mais plutôt un corpus de méthodes adaptées à des problèmes particuliers [Judistky et al., 1995, Sjöberg et al., 1995].

Les approches basées sur les noyaux pour la modélisation non linéaire ont attiré de plus en plus l'attention de la communauté de l'identification ces dernières années [Pillonetto et al., 2014]. À ce jour, la plupart des recherches sur le sujet en matière de contrôle et d'identification se sont concentrées sur l'un ou l'autre des deux paradigmes distincts, provenant respectivement des statistiques et des communautés d'apprentissage automatique : les *Processus Gaussian* et les *Machines Vectorielles de Soutien par les Moindres-Carrés*.

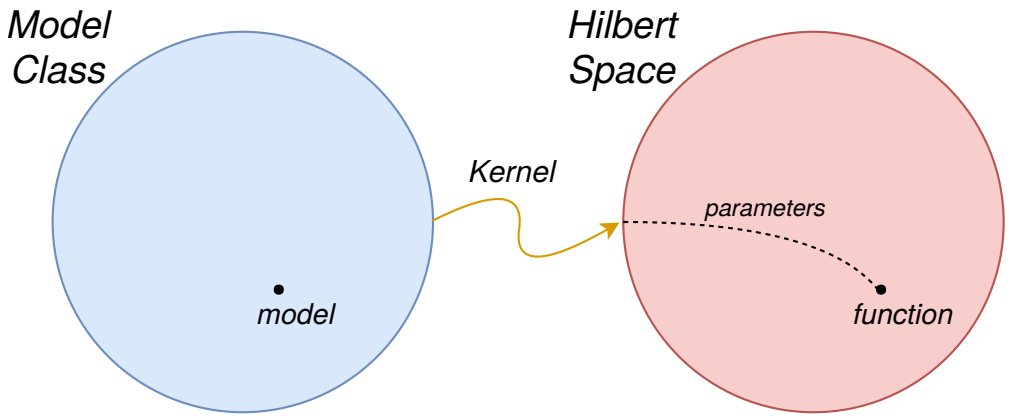



FIGURE 1 – *Identification dans des espaces Hilbert à noyau reproduisant : le choix du* $\mathbf{K}$ *détermine les caractéristiques de le modèle estimé.*

Une interprétation du *processus gaussien* [Rasmussen and Williams, 2006] relie la fonction du noyau à la fonction de covariance d'un processus aléatoire stochastique, permettant ainsi une interprétation probabiliste du problème.

En revanche, l'approche des *machines vectorielles de soutien par les moindres carrés* [Suykens et al., 2002] fournit un cadre déterministe pour l'estimation, considérant le noyau comme un mappage entre un espace primal (entrée) - dans lequel le problème d'apprentissage est formulé - et un espace (caractéristique) double dans lequel le problème est résolu.

Les deux fournissent un cadre puissant pour l'estimation de modèles non linéaires, et ont été appliqués avec succès à de nombreux problèmes d'un intérêt particulier pour l'identification de systèmes, e.g. [Goethals et al., 2005a, Tóth et al., 2011, Laurain et al., 2015, Darwish et al., 2015].

Récemment cependant, l'attention a également commencé à se tourner vers d'autres approches basées sur le noyau, commes *les Espaces de Hilbert à Noyau Reproduisant (RKHS)* [Aronszajn, 1950]. Les méthodes *RKHS* fournissent un lien conceptuel entre les approches populaires du noyau et l'identification classique du système, en reliant directement des notions fondamentales dans l'identification non linéaire (tels que la classe du modèle, la structure du modèle et la complexité) à la spécification d'un espace de caractéristiques non linéaires de haute dimension $\mathcal{H}$ à travers la définition d'une fonction noyau $\mathbf{K}$ (e.g. Figure 1).

Les modèles sont formés en prenant des combinaisons linéaires de noyaux pondérés, situés dans tout l'espace d'entrée.

$$\mathcal{M}: \ f(x) = \sum_i \alpha_i k_{x_i}(x), \ \alpha_i \in \mathbb{R}. \tag{1}$$

De cette facon, en définissant différentes fonctions du noyau (dont beaucoup sont facilement disponibles dans la littérature [Poggio and Girosi, 1990, Rasmussen and Williams, 2006]), le même ensemble de techniques peut être directement transposé à une gamme de problèmes différents. De tels problèmes sont intrinsèquement *mal posés*, et nécessitent des contraintes supplémentaires sur le comportement de la fonction (i.e. des termes de regularisation) pour assurer l'unicité de la solution, e.g.

$$\mathcal{J}(f) = \sum_{k=1}^{N} (y_k - f(x_k))^2 + \lambda \|f\|_{\mathcal{H}}^2, \tag{2}$$

où $\lambda$ est utilisé pour contrôler le compromis biais-variance du modèle.

Une autre approche de plus en plus populaire de l'identification basée sur le noyau s'appuie sur la méthodologie *RKHS* en considérant la formulation d'un modèle dans *un espace de Sobolev* [Adams, 1975]. Les méthodes Sobolev (e.g. smoothing splines) sont apparues ces dernières années comme un autre cadre mathématiquement élégant et efficace sur le plan des calculs pour les problèmes d'identification non linéaire [Wahba, 1990, Pillonetto et al., 2014].

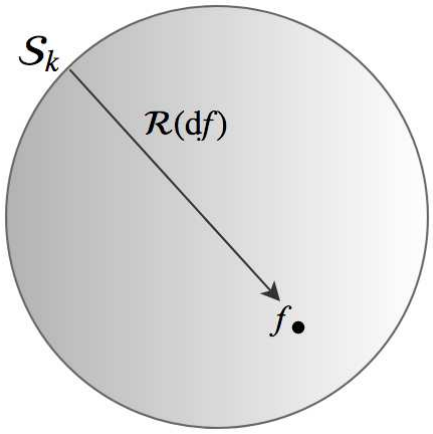



FIGURE 2 – *Dans un espace Sobolev, un noyau flexible est definé avec des contraints* ***souples*** *appliqués à travers d'un terme de regularisation.*

Ils permettent l'incorporation directe des contraintes structurelles (en particulier le lissage) dans le critère d'optimisation en codant les dérivées fonctionnelles dans la norme de l'espace des caractéristiques, e.g.

$$\|f\|_{\mathcal{S}_k^p} = \sum_{i=0}^{k} \int_{\mathcal{X}} \left( \frac{d^i f(x)}{dx^i} \right)^p dx. \tag{3}$$

Pour $p = 2$, $\mathcal{S}_k^2$ est un espace Hilbert aditionellement défini par ses dérivés fonctionnels. Formuler le problème de cette manière a de nombreuses implications, notamment :

— Bien que la définition d'un espace Sobolev, la définition du $\mathbf{K}$ est implicite. Par conséquent, le choix du noyau n'est pas gratuit, mais le noyau optimal pour chaque problème doit être déterminé. Il en résulte une fonction noyau a priori flexible, sans hyperparamètre.
— Dans ce cadre, dans (2), $\lambda$ contrôle non seulement la variance du modèle, mais aussi la mesure dans laquelle les propriétés définies en (2.16) sont respectées dans le modèle. Ainsi, les contraintes *douces* sur les propriétés du modèle peuvent être contrôlées en continu à travers un hyperparamètre de régularisation - par opposition aux contraintes *dures* placées par le choix de la fonction noyau (comme illustré dans la Figure 2).

Cependant, ils manquent de l'adaptabilité des approches RKHS conventionnelles. Comme la fonction du noyau est déterminée en résolvant la fonction du Green correspondant au problème, la détermination de nouvelles solutions n'est pas triviale dans la pratique - dans certains cas même infaisable.Par conséquent, l'utilisateur est intrinsèquement limité : à la fois dans la facon dont cette méthode peut être facilement appliquée à différents problèmes non linéaires et dans quels types de propriétés structurelles peuvent être pénalisées.

La contribution majeure de cette thèse est le développement d'une approche de régularisation dérivée cohérente avec le paramètre RKHS traditionnel : permettre des choix structurels *durs* à travers l'optimisation du noyau et des choix structurels *doux* à travers un terme de régularisation (comme dans la figure 3).

En formulant et en pénalisant directement les dérivées dans l'espace des caractéristiques choisi [Zhou, 2008, Bhujwalla et al., 2016a, Bhujwalla et al., 2016b, Bhujwalla et al., 2017b, Lauer et al., 2012] :

$$\mathcal{J}(f) = \sum_{k=1}^{N} (y_k - f(x_k))^2 + \lambda \left\| \frac{d^m f(x)}{dx^m} \right\|_{\mathcal{H}}^2, \tag{4}$$

une approche peut être facilement mise en œuvre dans des scénarios pratiques d'identification non linéaire. Une telle approche peut permettre : l'approximation de structures non linéaires, la détection structurelle, l'optimisation de la complexité, la réduction de la variance et la simplification du problème d'optimisation de l'hyperparamètre.

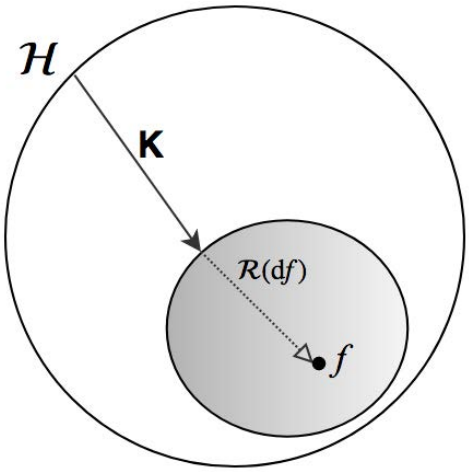



FIGURE 3 – *Le formulation des contraints aux derivés dans un RKHS directement permette l'optimisation* ***dur*** *et* ***souple*** *de la classe des modèles.*

Cette méthode sera largement étudiée dans la thèse :

- Un cadre général pour la régularisation des dérivés dans l'identification non linéaire sera développé.
- Les préoccupations théoriques et pratiques seront abordées.
- Une comparaison entre les approches développées et les méthodes de la littérature sera présentée.
- Et la méthode sera appliquée à plusieurs problèmes d'intérêt particulier dans l'identification non linéaire.

En plus des exemples de simulation, une application des méthodes proposées à une étude de cas industrielle sera également présentée. En collaboration avec *Post Luxembourg*, un fournisseur d'accès Internet de niveau 2 et un réseau de diffusion de contenu, un cadre a été développé pour la modélisation et la prévision du trafic provenant du réseau et provenant de différents systèmes autonomes [Bhujwalla et al., 2017a].

## Panorama

Cette thèse se concentrera exclusivement sur l'application de méthodes non paramétriques basées sur le noyau à des problèmes d'identification non-linéaires.

Comme pour les autres méthodes non linéaires, deux questions clés dans l'identification basée sur le noyau sont les questions de *comment définir un modèle non linéaire* (sélection du noyau) et *comment ajuster la complexité du modèle* (régularisation).

La contribution principale de cette thèse est la présentation et l'étude de deux critères d'optimisation (un existant dans la littérature et une nouvelle proposition) pour l'approximation structurale et l'accord de complexité dans l'identification de systèmes non linéaires basés sur le noyau. Les deux méthodes sont basées sur l'idée d'intégrer des contraintes de complexité basées sur des caractéristiques dans le critère d'optimisation, en pénalisant les dérivées de fonctions. Essentiellement, de telles méthodes offrent à l'utilisateur une certaine souplesse dans la définition d'une fonction noyau et dans le choix du terme de régularisation, ce qui ouvre de nouvelles possibilités quant à la facon dont les modèles non linéaires peuvent être estimés dans la pratique.

Les deux méthodes ont des liens étroits avec d'autres méthodes de la littérature, qui seront examinées en détail dans les chapitres 2 et 3 et formeront la base des développements ultérieurs de la thèse. Alors que l'analogie sera faite avec des cadres parallèles, la discussion sera ancrée dans le cadre de *Reproducing Kernel Hilbert Spaces* (RKHS). L'utilisation des méthodes RKHS permettra d'analyser les méthodes présentées d'un point de vue à la fois théorique et pratique.

De plus, les méthodes développées seront appliquées à plusieurs «études de cas» d'identification, comprenant à la fois des exemples de simulation et de données réelles, notamment:

- Détection structurelle dans les systèmes statiques non linéaires.
- Contrôle de la fluidité dans les modèles LPV.
- Ajustement de la complexité à l'aide de pénalités structurelles dans les systèmes NARX.
- Modelisation de trafic internet par l'utilisation des méthodes à noyau.

## Structure de la Thèse

La thèse sera structurée comme suit :

- *Chapitre 1: Introduction*

  Dans ce chapitre, un bref apercu non technique de l'identification des systèmes non linéaires sera présenté, dans le but de familiariser le lecteur avec le sujet et d'introduire les sujets abordés dans les chapitres suivants de cette thèse.

- *Chapitre 2: Des Noyaux Reproduisant dans l'Identification des Systèmes Nonlinéaires*

  Le chapitre suivant présentera l'état de l'art actuel en ce qui concerne les méthodes du noyau dans l'identification de systèmes non linéaires, en mettant l'accent sur deux approches : les méthodes RKHS et *Sobolev*. La discussion sera centrée sur la question du choix d'une fonction noyau, et en particulier, comment cela se rapporte à la définition de classe de modèle dans un cadre d'identification classique.

- *Chapitre 3: Regularisation des Derivés dans des Espaces Hilbert au Noyau Reproduisant*

  Dans le chapitre trois, la discussion se tournera vers les développements de cette thèse, c'est-à-dire la formulation de méthodes pour pénaliser les dérivés dans un RKHS. Deux méthodes seront présentées, une issue de la littérature et une proposition originale. Chaque schéma d'optimisation sera formulé avec soin, en accordant une attention aux considérations théoriques et pratiques.
  En conclusion, un exemple comparatif sera présenté en utilisant toutes les méthodes formulées jusqu'à présent. Le but de cet exemple sera d'illustrer la théorie discutée, et de fournir une comparaison objective entre la performance des méthodes présentées dans les chapitres deux et trois.

- *Chapitre 4: Études de Cas*

  Au chapitre quatre, les méthodes présentées dans le chapitre précédent seront appliquées à plusieurs exemples de simulation, dans le but de montrer comment elles peuvent être utilisées dans la pratique. Trois scénarios distincts seront considérés, chaque exemple étant sélectionné pour illustrer un aspect différent des méthodes présentées :
  · Détection structurelle dans les systèmes statiques non linéaires.
  — On montrera que l'utilisation d'une pénalité dérivée offre un moyen de déconstruire la définition de la classe du modèle dans les petits problèmes, offrant une meilleure

précision et une meilleure compréhension physique de certains types de problèmes.
  - Contrôle de la fluidité dans les modèles LPV.
    - Pour illustrer comment les pénalités dérivées peuvent être appliquées aux types de structures de modèles dynamiques d'intérêt dans les communautés de contrôle et d'identification, les méthodes présentées seront formulées pour le cas des modèles LPV.
  - Ajustement de la complexité à l'aide de pénalités structurelles dans les systèmes NARX.
    - On montrera également comment des pénalités dérivées peuvent être utilisées pour formuler des contraintes structurelles, offrant une toute nouvelle facon d'agir sur les propriétés d'un modèle. À la connaissance de l'auteur, il y a peu ou pas de discussion de ces méthodes dans la littérature, car il est extrêmement difficile d'obtenir un contrôle similaire sur les propriétés du modèle dans d'autres cadres basés sur le noyau.
    - Des contraintes structurelles pourraient présenter un intérêt à bien des égards, par ex. détection ou approximation structurelle et sélection de variables. Nous ne considérerons ici qu'une seule application possible, à savoir comment des contraintes structurelles peuvent être utilisées en plus des techniques existantes, offrant un degré de liberté supplémentaire dans le problème d'optimisation et fournissant ainsi un moyen simple d'améliorer les performances dans la modélisation non linéaire.

- *Chapitre 5: Application aux Données*

  La dernière partie de cette thèse portera sur une application à un problème de données réelles : la prévision du trafic réseau pour le diagnostic et la maintenance du réseau.
  Ce chapitre documente la première partie d'une collaboration continue avec un partenaire industriel, et discutera du besoin de modèles et des défis impliqués dans la modélisation des données de réseau.
  On montrera qu'une approche basée sur le noyau fournit une approche précise, flexible et conviviale pour de tels problèmes, et comment la régularisation dérivée peut être utilisée dans ce contexte pour simplifier le problème de modélisation.

## Publications Associées

1. *The Impact of Smoothness on Model Class Selection in Nonlinear System Identification: An Application of Derivatives in the RKHS*
   Y. Bhujwalla, V. Laurain et M. Gilson
   Présenté au 2016 American Control Conference (ACC), Boston, USA.
2. *An RKHS Approach to Systematic Kernel Selection in Nonlinear System Identification*
   Y. Bhujwalla, V. Laurain et M. Gilson
   Présenté au 2016 IEEE Conference on Decision and Control (CDC), Las Vegas, USA.
3. *An AS-Level Approach to Network Traffic Analysis and Modelling*
   Q. Grandemange, Y. Bhujwalla, M. Gilson, O. Ferveur et E. Gnaedinger
   Présenté au 2017 IEEE International Conference on Communications (ICC), Paris, France.
4. *How We Spend Our Time Online: Predicting Network Traffic Using System Identification*
   Y. Bhujwalla, Q. Grandemange, M. Gilson, V. Laurain et E. Gnaedinger
   Présenté au 2017 IFAC World Congress, Toulouse, France.

5. *An RKHS Approach to Controlling Smoothness in Nonparametric LPV-IO Identification*
   Y. Bhujwalla, V. Laurain et M. Gilson
   Présenté au 2017 IFAC World Congress, Toulouse, France.

*Mot clés : identification des systèmes, méthodes à noyaux, regularisation, espaces Hilbert à noyau reproduisant, derivés des fonctions dans les espaces Hilbert.*

# Liste des notations, symboles et abréviations

### *Notations et symboles*

| | | |
|---|---|---|
| $a$ | : | scalar, $a \in \mathbb{R}$ |
| $\mathbf{a}$ | : | vector, $\mathbf{a} = \begin{bmatrix} a_1 & a_2 & \cdots & a_{n_a} \end{bmatrix}^\top \in \mathbb{R}^{n_a}$ |
| $\mathbf{A}$ | : | matrix, $\{\mathbf{A}\}_{i,j} = a_{i,j} \in \mathbb{R}$ |
| $u_k$ | : | *input variable at time* $k$, $u_k \in \mathbb{R}^{n_u}$ |
| $p_k$ | : | *scheduling variable at time* $k$, $p_k \in \mathbb{P} \subset \mathbb{R}^{n_p}$ |
| $y_k$ | : | *output variable at time* $k$ |
| $y_{\mathrm{o},k}$ | : | *noise-free output variable at time* $k$ |
| $e_{\mathrm{o},k}$ | : | *noise disturbance at time* $k$ |
| $\mathbf{x}_k$ | : | *regressor variable at time* $k$, $\mathbf{x}_k = \begin{bmatrix} x_{1,k} & x_{2,k} & \cdots & x_{n_x,k} \end{bmatrix}^\top \in \mathbb{R}^{n_x}$ |
| $\mathcal{X}$ | : | *input space* $\mathcal{X} \subset \mathbb{R}^{n_x}$ |
| $f_{\mathrm{o}}(\cdot)$ | : | *system mapping* $f_{\mathrm{o}} : \mathcal{X} \to \mathbb{R}$ |
| $\mathcal{S}_{\mathrm{o}}$ | : | *system of interest* |
| $\mathcal{Z}_n$ | : | *measured data* |
| $\mathcal{M}$ | : | *estimation model class* |
| $\mathcal{M}_{\mathrm{ALG}}$ | : | *modelling framework corresponding to* ALG *algorithm* |
| $\| \cdot \|_{\mathcal{V}}$ | : | *norm in functional space* $\mathcal{V}$ |
| $\langle \cdot, \cdot \rangle_{\mathcal{V}}$ | : | *inner-product in functional space* $\mathcal{V}$ |
| $f \in \mathcal{V}$ | : | *function in functional space* $\mathcal{V}$ |
| $\hat{f} \in \mathcal{V}$ | : | *optimised function in functional space* $\mathcal{V}$ |
| $\mathcal{H}$ | : | *Hilbert Space* |
| $\mathcal{H}_k$ | : | *Reproducing Kernel Hilbert Space* |
| $\mathcal{S}_m^p$ | : | *p-norm* $m^{th}$*-order Sobolev Space* |
| $k_{x_i}(x)$ | : | *kernel slice of* $x_i$ *evaluated at* $x$ |
| $k_x$ | : | *kernel slices* $k_x = \{k_{x_1}(x), k_{x_2}(x), k_{x_3}(x) \ldots\}$ *evaluated at* $x$ |
| $\mathrm{K}(x_i, x_j)$ | : | *reproducing kernel of* $x_i$ *and* $x_j$ |
| $\mathbf{K}$ | : | *kernel matrix* $\{\mathbf{K}\}_{i,j} = \mathrm{K}(x_i, x_j) \in \mathbb{R}$ |
| $\alpha, \beta$ | : | *model parameters corresponding to a nonparametric, nonlinear function* |
| $\theta$ | : | *model parameters corresponding to a parametric, linear function* |

| | | |
|---|---|---|
| $\sigma, \gamma$ | : | *kernel hyperparameters* |
| $\lambda$ | : | *regularisation hyperparameter* |
| $\mathcal{J}(\cdot)$ | : | *optimisation criteria (or cost-function)* |
| $\mathcal{R}(\cdot)$ | : | *regularisation term* |
| $\mathcal{F}$ | : | *nonparametric model definition (representer)* |
| $\partial_x^m(\cdot)$ | : | *$m^{th}$-order partial derivative wrt. $x$* |
| $\mathcal{D}^{\mathbf{m}}(\cdot)$ | : | *differential operator* |
| $\rho_k$ | : | *kernel density* |
| $\Delta_x$ | : | *maximum spacing of observations in $\mathcal{X}$* |
| $\epsilon_{\hat{f}}$ | : | *smoothness tolerance parameter* |

### *Abréviations*

| | | |
|---|---|---|
| FIR | : | *finite impulse response* |
| ARX | : | *autoregressive with exogenous input* |
| ARMAX | : | *autoregressive moving average* |
| OE | : | *output-error* |
| BJ | : | *Box-and-Jenkins* |
| LTI | : | *linear time-invariant* |
| LPV | : | *linear parameter-varying* |
| NARX | : | *nonlinear autoregressive with exogenous input* |
| ANARX | : | *additive nonlinear autoregressive with exogenous input* |
| RKHS | : | *Reproducing Kernel Hilbert Space* |
| FIT | : | *best fit rate* |
| MBIAS | : | *mean bias (over input space)* |
| MSDEV | : | *mean standard deviation (over input space)* |
| MRMSE | : | *mean root mean-squared error (over Monte-Carlo trials)* |
| F-REG | : | *functional-norm regularisation in the RKHS* |
| SPLINES | : | *functional-norm regularisation in the Sobolev Space* |
| D-MIN | : | *penalisation of derivative evaluations in the RKHS* |
| D-REG | : | *derivative regularisation in the RKHS* |
| RBF | : | *radial basis function* |
| AS | : | *autonomous system* |
| ISP | : | *internet service provider* |
| CDN | : | *content delivery network* |
| QOE | : | *quality of experience* |
| IXP | : | *internet exchange point* |
| DHR | : | *dynamic harmonic regression* |

# Table des figures

# Liste des tableaux

# 1

# Introduction

*In this chapter a brief, non-technical overview of nonlinear system identification will be presented, with the aim of familiarising the reader with the subject and introducing the topics addressed in the subsequent chapters of this thesis.*

## 1.1 System Identification

*System identification* is concerned with the statistical modelling of systems from data, for application in control engineering [Åström and Eykhoff, 1971, Söderström and Stoica, 1988, Gevers, 2006, Ljung, 1999, Young, 2011, Pintelon and Schoukens, 2012].

In this context, the *system* is the entity to be studied, which could be any combination of physical or non-physical processes as required. The notion of a *model* refers to an abstraction of the system, whereby the behaviour of the process or system of interest is approximated by mathematical expressions (Figure 1.1).

System identification is focused on *data-driven* modelling, as opposed to the development of models from physical laws (also termed *first-principles modelling, mathematical modelling* and *white-box modelling*).

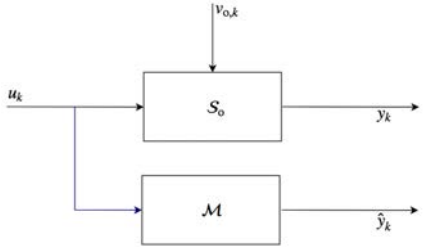



FIGURE 1.1 – Estimating a model $\mathcal{M}$ from observations at the input $u_k$ and output $y_k$ of an unknown system $\mathcal{S}_o$, subject to disturbances $v_{o,k}$.

Constructing models either solely from experimental data (*black-box modelling*) or through a combination of data and physical knowledge (*grey-box modelling*) is often of interest in practical engineering scenarios. First-principles modelling is often both time-consuming and challenging,

requiring extensive knowledge of the underlying physical system, and can lead to highly complex models, unsuitable for application in control [Garnier and Wang, 2008]. A data-driven approach affords the user a convenient method of developing accurate models, suited to the requirements of the user.

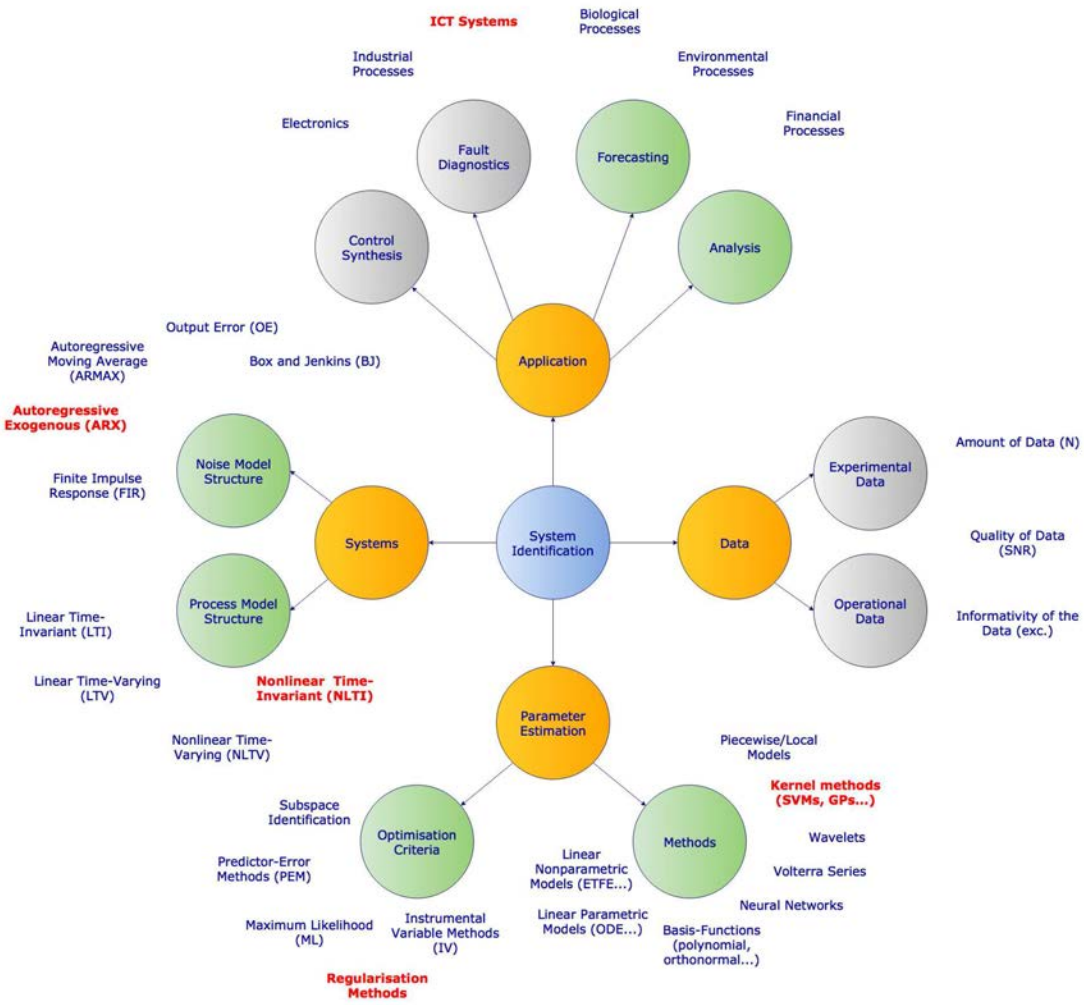



FIGURE 1.2 – A *nonexhaustive* summary of popular research topics in system identification (subjects addressed in this thesis highlighted in red).

System identification is therefore underpinned by statistical theory in a similar manner to other branches of data-driven modelling such as machine learning, econometrics and signal processing. Though many of these fields frequently overlap both in their theoretical formulation [Parzen, 1960, Vapnik, 1998] and increasingly in their application [Ljung, 2010, Hall et al., 2012], there are certain topics of particular interest that broadly allow us to classify the domain of system identification (as illustrated in Figure 1.2) notably:

- **Application-Oriented Modelling:**
  One of the fundamental tenets of statistics is summarised by the well-known statement *"all models are wrong, some are useful"* [Box, 1976]. Useful - or *fit for purpose* - could mean that a model provides physical insight, or that it functions as an effective predictor

over a horizon of interest. But in system identification, one implication of this statement is that the usefulness of a model is not only determined by its accuracy but also by its suitability for its intended purpose, e.g. in control synthesis or fault diagnosis.

- **Causal Systems:**
  Engineering systems are typically causal, in that outputs (denoted as $y$) often depend on a set of inputs or control variables signals (denoted as $u$).
- **Dynamical Systems:**
  Industrial processes are physical systems and therefore typically dynamical, in that the state of a system at a particular moment $k$ depends not only on the input to the system at that moment ($u_k$) but also past states of the system ($y_{k-1}$, $y_{k-2}$, ...).
- **Experimental Data:**
  In system identification, measurements may be either operational data or experimental data. In the case of the latter, the design of reliable experiments (encompassing subjects such as measurement theory, instrumentation and input signal design) then naturally becomes an important area of research within the subject [Bombois et al., 2006, Gevers et al., 2008, Valenzuela et al., 2015].
- **Noise Modelling:**
  Understanding how disturbances influence the behaviour is a crucial area of system identification (where *disturbances* could be unmeasured phenomena, uncontrollable signals or stochastic perturbations). The understanding of both noise structures (where disturbances enter the system) and noise models (how they affect the system) is defining aspect of system identification [Box et al., 1970].

It is not that any of the above topics are in themselves the exclusive preserve of system identification, but rather that a combination of these issues is a common feature of identification problems.

By way of comparison, we consider the example of financial modelling [Wooldridge, 2015, Stock and Watson, 2003], e.g. the prediction of share prices in a marketplace. Whilst identification methods could be applied to such a problem, more often methods from the econometrics and signal processing communities are used.

Nonetheless, financial modelling still bears some resemblance to identification problems: systems are also dynamical, in that the value of an asset tomorrow will depend upon its value today; and noisy - value is subject to disturbances in the form of uncontrollable market forces (e.g. speculation, competition and government policy).

However, models of financial systems are typically not developed from experimental data, as such data can rarely be obtained. They are usually modelled as *time-series* (or input-free) systems, as input variables are often either unknown, unmeasurable or uncontrollable. And such models are usually used for one of either two purposes: forecasting and analysis. Hence, there are arguably fewer constraints on the model structure as their suitability for control and diagnostics is not an issue, and accuracy is the major modelling objective.

By contrast, aerospace applications (such as modelling the dynamics of a plane, helicopter or satellite) are very well-suited to system identification methods. Systems are dynamical: aircraft are physical systems, and their trajectories are governed by the laws of motion; and noisy: disturbances are present both in the system (process noise) and at the output (measurement noise). Furthermore, the modelling objective is usually to understand how the vessel responds to a particular set of stimuli, e.g. the angular velocity and pitch of a rotor. Hence, it is imperative

to find a safe, efficient and informative procedure for testing (experiment design). And, whilst estimated models *should* be as accurate *as possible*, they *must* be compatible with any envisaged controller.

And in fact, many of the important advances in system identification can be seen to have emerged from a vibrant research community faced with a unique set of problems [Gevers, 2006, Ljung, 2010] (including, but not limited to continuous-time identification [Garnier and Wang, 2008], closed-loop identification [Van Den Hof and Schrama, 1995, Gilson and Van Den Hof, 2005], frequency domain identification [Pintelon and Schoukens, 2012], subspace methods [Van Overschee and De Moor, 2012], block-oriented identification [Giri and Bai, 2010], LPV identification [Tóth, 2010] and nonlinear identification [Nelles, 2001, Billings, 2013]).

Note that the above list in no way constitues an exhaustive list of the developments that have been made in the field, but merely a broad statement of some of the challenges often faced in a modelling scenario.

Furthermore, whilst system identification has historically been defined by its application to industrial systems, in recent years its scope has expanded much beyond this. There are many examples of identification methods being applied to a range of areas outside of the traditional industrial applications, such as environmental systems [Gilson et al., 2012, Laurain et al., 2014] and biological systems [Kulkarni et al., 2014, Dalchau, 2012].

Hence, system identification is a subject that both owes much and has contributed much to the statistical modelling community.

## 1.2 Nonlinear System Identification

In system identification, our objective is to construct a model $\mathcal{M}$, given input-output data obtained from a system of interest $\mathcal{S}_o$

$$\mathcal{Z}_N = \{(u_1, y_1), (u_2, y_2), \ldots, (u_N, y_N)\}, \tag{1.1}$$

where $u_k \in \mathbb{R}^{n_u}$ and $y_k \in \mathbb{R}$ denote the input and output of the system at time $k$. This section will discuss some the challenges facing an engineer in typical *nonlinear* identification scenarios.

Although many naturally-occurring and industrial systems are well-approximated by linear models, in reality extremely few systems are perfectly described by linear models. Most systems are to some extent nonlinear. And although linear approximations are often satisfactory, in many cases they are insufficient for their desired purpose, leading to a natural requirement for nonlinear system identification methods, such as those developed in [Nelles, 2001, Billings, 1980, Sjöberg et al., 1995, Judistky et al., 1995].

A function $f(x)$ can be described as linear if it adheres to the *principle of superposition:*

$$\begin{aligned} \textbf{additivity} \quad &\Rightarrow \quad f(x_1 + x_2) \quad = f(x_1) + f(x_2) \\ \textbf{homogeneity} \quad &\Rightarrow \quad f(cx) \qquad = c f(x), \end{aligned} \tag{1.2}$$

where $x, x_1, x_2$ are inputs to the system and $c \in \mathbb{R}$ is a scalar. For a *linear time-invariant autoregressive systems with exogenous inputs* (LTI-ARX), this corresponds to a structure of the

following form:

$$\mathcal{S}_{lin,arx} : \begin{cases} y_{o,k} & = \sum_{i=1}^{n_x} \theta_{o,i} x_{i,k} \\ y_k & = y_{o,k} + e_{o,k}. \end{cases} \tag{1.3}$$

In the above expression, $\theta_o \in \mathbb{R}^{n_x}$ are parameters characterising the behaviour of the system, $y_{o,k} \in \mathbb{R}$ is the noise-free output of the system, $e_{o,k} \sim \mathcal{N}(0, \sigma_e^2)$ is a white Gaussian noise disturbance and $\mathbf{x_k} \in \mathcal{X}$ is the *regressor vector*, composed of the past inputs and outputs of $\mathcal{S}_o$:

$$\mathbf{x_k} = \left[ y_{k-1} \cdots y_{k-n_a} \quad u_{1,k} \cdots u_{1,k-n_b} \quad u_{2,k} \cdots u_{n_u,k-n_b} \right]^\top \in \mathcal{X} \tag{1.4}$$

at each instant $k \in [1 \ldots N]$, where the input space $\mathcal{X}$ is a compact (and non-empty) set $\mathcal{X} \subset \mathbb{R}^{n_x}$ for $n_x = n_a + n_u (n_b + 1)$.

Hence nonlinearity is a *non-property*, in that it describes any system not exhibiting the properties described in (1.2). For ARX systems, this corresponds to any system which cannot be described by the definition of (1.3). Clearly then, formulating a generalised framework for the identification of nonlinear systems is very difficult, as to state that a system is nonlinear is not in itself an informative statement.

But what does this uninformative statement correspond to ? Before proceeding further, a generalised formulation for a particular class of nonlinear systems (NARX systems) will be presented. This will form the basis for the discussion of both this chapter, and the methods presented in the subsequent chapters.

### 1.2.1 Problem Statement

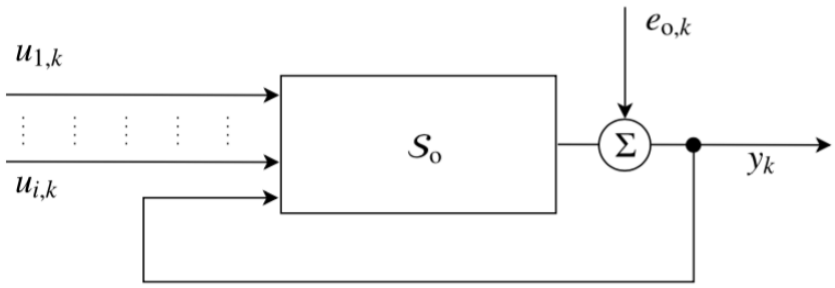



FIGURE 1.3 – An example NARX system to be identified

Whilst it will not be assumed that the unknown system is linear, we will nonetheless make several strong assumptions throughout the thesis :

A1: All examples considered in this thesis will be types of *nonlinear autoregressive systems with exogenous inputs* (NARX systems), as in Figure 1.3. In an ARX model, disturbances are assumed to be due to process noise (as opposed to measurement noise) and white Gaussian in nature. Investigation of nonlinear identification problems under more general conditions, better reflecting most industrial applications can be found in the literature, for example [Billings and Chen, 1989, Laurain et al., 2015].

A2: All examples considered will be systems of known input delay (and hence neglected for convenience) and known order ($n_a$ and $n_b$ in (1.4)). Both delay and order estimation in

nonlinear systems remain challenging issues, but are studied to a certain extent in the literature, e.g. [Ljung, 1999, Nelles, 2001].

A3: The discussion will be restricted to nonlinear *time-invariant* systems, that is *if* $\|\mathbf{x}_{k_1} - \mathbf{x}_{k_2}\| = 0$, for any $k_1 \neq k_2$ in (1.5), then $y_{\mathrm{o},k_1} - y_{\mathrm{o},k_2} = 0$ is also true. The identification of nonlinear time-varying systems is beyond the scope of this thesis.

A4: All developments in this thesis will be presented for the case of *multiple-input single-output* (MISO) systems. Certain examples will be restricted to the case of *single-input single-output* (SISO) systems.

Using the statements above, $\mathcal{S}_{\mathrm{o}}$ can be described using the following relations :

$$\mathcal{S}_{nl,arx} : \left\{ \begin{array}{ll} y_{\mathrm{o},k} & = f_{\mathrm{o}}(\mathbf{x_k}) \\ y_k & = y_{\mathrm{o},k} + e_{\mathrm{o},k}. \end{array} \right. \tag{1.5}$$

In (1.5), $f_{\mathrm{o}} : \mathcal{X} \to \mathbb{R}$ is no longer a set of parameters, but rather an unknown nonlinear function. Hence, identifying $\mathcal{S}_{\mathrm{o}}$ is equivalent to characterising the function $f_{\mathrm{o}}$.

### 1.2.2 Nonlinear Model Structures

Between the LTI-ARX model $y_{\mathrm{o},k} = \sum_{i=1}^{n_x} \theta_{\mathrm{o},i} x_{i,k}$ (1.3) and the NARX model $y_{\mathrm{o},k} = f_{\mathrm{o}}(\mathbf{x_k})$ (1.5), clearly there is quite a bit of play. In (1.5), not only is the *type* of nonlinearity undefined (e.g. harmonic, saturation, hysteresis), but also the way in which variables and functions of variables interact.

Beyond very trivial cases, there are simply too many degrees of freedom in the problem to be able to efficiently estimate useful nonlinear models, without making some assumptions regarding the behaviour of the unknown system. Hence, to date, most research in nonlinear identification has focused on the investigation of 'subproblems' (i.e. classes of nonlinear systems) of particular interest.

These could be physically meaningful structures, i.e. models that provide a good approximation of many real-world nonlinearities. Or, control-relevant structures, i.e. structures that offer greater flexibility than an LTI model, but are still suitable for control synthesis. Or even, structures that are primarily interesting from a statistical modelling perspective. In all of these cases, constraints are placed on how nonlinearities enter into the system, and how variables interact with other components of the system.

**Example 1: Block Models**

Whilst block oriented models will not be considered in this thesis, they are an important and well-studied class of nonlinear systems [Billings, 1980, Schoukens et al., 2003, Goethals et al., 2005a, Giri and Bai, 2010], as they offer a control-relevant framework capable of approximating many real-world nonlinearities.

In a block model structure, it is assumed that any nonlinearities present are static, but that they may be combined with dynamical linear components. For example, *Hammerstein* systems consider the presence of a static nonlinearity preceeding a dynamical linear component:

$$\mathcal{S}_{Hamm,arx} : \left\{ \begin{array}{ll} \tilde{u}_k & = f_{\mathrm{o}}(u_k) \\ \tilde{x}_k & = \left[ \, y_{k-1} \cdots y_{k-n_a} \; \tilde{u}_k^\top \right]^\top \in \mathbb{R}^{n_{\tilde{x}}} \\ y_{\mathrm{o},k} & = \sum_{i=1}^{n_{\tilde{x}}} \theta_i \tilde{x}_k(i), \quad \theta \in \mathbb{R}^{n_{\tilde{x}}} \\ y_k & = y_{\mathrm{o},k} + e_{\mathrm{o},k}. \end{array} \right. \tag{1.6}$$

*Wiener* systems consider the inverse (linear + nonlinear), and *Wiener-Hammerstein* systems consider combinations of the two (nonlinear + linear + nonlinear).

By combining linear and nonlinear 'blocks' in different ways, such classes of systems can be used to approximate a wide range of nonlinearities. Furthermore, in many cases block representations actually provide a very reasonable approximation of how nonlinearities (e.g. friction) enter into dynamical systems in practice.

**Example 2: LPV Models**

*Linear parameter-varying* (LPV) systems are another very important class of nonlinear systems, and one well-studied in the identification literature [Bamieh and Giarré, 2002, Lovera and Mercere, 2007, Tóth, 2010, Laurain et al., 2012].

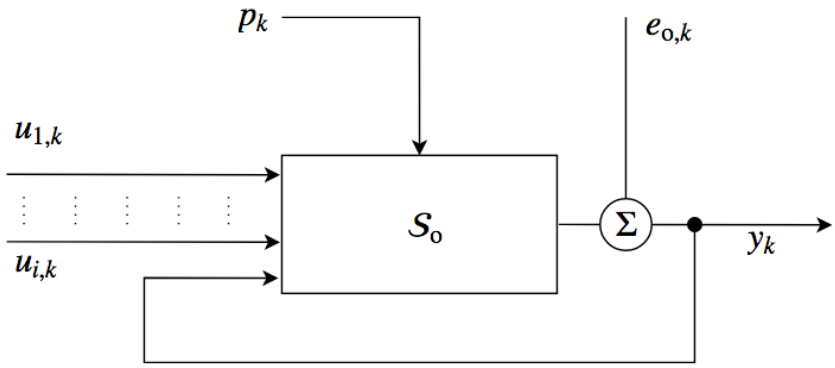



FIGURE 1.4 – An example *linear parameter-varying* (LPV) ARX system

In an LPV system, there is a linear relationship between the input and output variables - as in the LTI case. However, this linear relationship is itself dependent upon another variable, usually referred to as the *scheduling variable* $p_k$ (Figure 1.4).

$$\mathcal{Z}_{N,lpv} = \{(u_1, p_1, y_1), \ldots, (u_N, p_N, y_N)\}, \quad p_k \in \mathbb{P} \subset \mathbb{R}^{n_p}. \tag{1.7}$$

This allows an additional degree of freedom in the modelling problem, whilst preserving a *linear-like* model structure, suitable for control :

$$\mathcal{S}_{lpv,arx} : \begin{cases} y_{o,k} & = f_o(x_k, p_k) \\ & = \sum_{i=1}^{n_x} f_{o,i}(p_k)\, x_{i,k}, \qquad f_{o,i} : \mathbb{P} \to \mathbb{R} \\ y_k & = y_{o,k} + e_{o,k}. \end{cases} \tag{1.8}$$

The scheduling variable usually corresponds to a known, measurable input to the system, but one that is typically either not controllable (e.g. ambient temperature and pressure in a chemical or metallurgical process) or not control-relevant, for example the altitude of an aircraft. In this case, the dynamics of the vessel may depend on the altitude - which can be indirectly controlled by the pilot. However ideally it should be possible to stabilise and manoeuvre the vessel without varying the altitude any more than necessary.

Whilst it is rare that real system will be strictly 'LPV', such a structure is significantly more flexible than an LTI representation and often offers much greater accuracy, whilst still allowing the user to benefit from an established control theory.

The identification of LPV systems will be considered in this thesis, with the intention of making the link between the work presented here and areas of active research in the identification community.

**Example 3: Additive Models**

*Additive nonlinear* structures have received less attention in the identification community than other nonlinear structures, such as block models and LPV models. This can in part perhaps be explained by their lack of obvious physical analogy with many industrial problems, and the lack of a clear advantage with respect to other nonlinear structures in a control setting.

They have however received widespread interest in other statistical communties [Hastie and Tibshirani, 1990, Wahba, 1990, Doumpos et al., 2007, Duvenaud et al., 2011]. The reasons for this are easily understood: additive nonlinear models allow considerably greater flexibility with respect to linear models, by allowing nonlinear relations between inputs and outputs, whilst circumventing certain fundamental statistical issues, such as *the curse of dimensionality* [Bellman, 1957], by neglecting the influence of interactions between variables.

$$\mathcal{S}_{anl,arx} : \begin{cases} y_{\mathrm{o},k} & = \sum_{i=1}^{n_x} f_{\mathrm{o},i}(x_{i,k}) \qquad f_{\mathrm{o},i} : \mathbb{R} \to \mathbb{R} \\ y_k & = y_{\mathrm{o},k} + e_{\mathrm{o},k} \end{cases} \tag{1.9}$$

These attractive features have not passed unnoticed in the nonlinear identification community, with increasing attention devoted to this problem, both in the estimation of additive structures e.g. [Bai, 2005, Mu et al., 2017b] and the decoupling of nonlinear components into additive structures e.g. [Tiels and Schoukens, 2013, Dreesen et al., 2015].

In a nonlinear identification context, *the curse of dimensionality* implies that the number of observations required to efficiently estimate a model increases exponentially with the dimension of the identification problem (i.e. the regressor variable).

In identification, this is an important point - as it means that obtaining sufficient quantities of reliable and informative measurements from experimental data may often be infeasible in practice. This could be for one of many possible reasons. For example, the cost of running experiments could be prohibitive, sufficient downtime may be unavailable, sensors may not be sufficiently accurate or it may not be possible to sufficiently excite the system due to safety concerns. Note that this contrasts somewhat with areas such as machine learning and data mining - where *big data* and *sparsity* problems are now commonplace ; the challenge now being to find meaningful information from enormous quantities of largely uninteresting data.

Hence, in this thesis we will also look at how structural constraints (such as additivity) can be used to improve the estimation of nonlinear models.

### 1.2.3 Defining a Nonlinear Model

In the previous section, the problem of choosing a suitable nonlinear structure for estimation was discussed. Already, this represents a considerable amount of additional freedom in the problem with respect to the classical *linear time-invariant* (LTI) case. However, the difficulty doesn't end there.

Let's assume we've been given data $\mathcal{Z}_n$, a model structure (e.g. *Wiener-Hammerstein* or LPV), a choice of noise model (e.g. FIR, ARX, ARMAX, OE or BJ in open-loop) and a choice of repre-

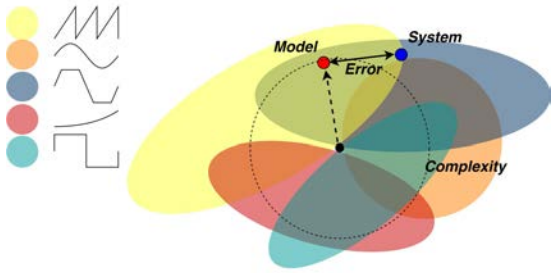



FIGURE 1.5 – In a *parametric* framework, the complexity of the model class is built-up until sufficient accuracy is ensured, and then subsequently acted upon to obtain a *parsimonious* representation.

sentation (discrete-time versus continuous-time ; time-domain versus frequency-domain ; input-output, state-space or series expansion).

To estimate a model, firstly we need to a define a candidate set of models $\mathcal{M}$ from which to choose a model (i.e. a *model class*). Then, given a candidate set, a suitable optimisation criteria (or cost-function) is required to estimate the model parameters in each case. Finally, a suitable model selection criteria is required to select the best model. This process is summarised in Figure 1.5. Whilst what constitutes the best model in any given scenario will always depend somewhat upon both modelling objectives and the selection criteria used, in general the aim is to find the *simplest* (or least complex) model that accurately reproduces the data. This is often referred to in the literature as the *parsimony principle* or *Occam's Razor* (see for example [Rasmussen and Ghahramani, 2001] for a discussion on the role of Occam's Razor in statistical modelling).

In the LTI case as per (1.3), a model is defined according to:

$$\mathcal{M}_{lin}: \quad f(\mathbf{x}_k) = \sum_{i=1}^{n_x} \theta_i x_{i,k}, \quad \theta \in \mathbb{R}^{n_x}. \tag{1.10}$$

Here, the model class - i.e. all the possible models that could be formulated using $\mathcal{M}_{lin}$ - is the span of $\theta$. The *complexity* of the model is its order, i.e. the number of parameters $n_x$. And the goal is to find the smallest value of $n_x$ such that good values of $\theta$ can be estimated.

### Parametric Methods

In the nonlinear case, both defining a model and evaluating its properties become more difficult. Consider a model of the following form, for $\mathcal{X} = \mathbb{R}$:

$$\mathcal{M}_{nl,par}: \quad f(x_k) = \sum_{i=1}^{n_\theta} \alpha_i \, \phi(x_k, \beta_i), \quad \alpha, \beta \in \mathbb{R}^{n_\theta}. \tag{1.11}$$

The model $\mathcal{M}_{nl,par}$ is a sum of weighted nonlinear mappings $\phi(\,\cdot\,, \beta_i) : \mathcal{X} \to \mathbb{R}$, with the values of $\alpha_i$ representing the weights and $\beta_i$ denoting some additional characterisation of $\phi$. The simplest example of this is perhaps a *polynomial* representation, where the mappings are monomials of order $\beta_i$:

$$\phi(x_k, \beta_i) = x_k^{\beta_i}, \quad \beta_i \in \mathbb{N}_0. \tag{1.12}$$

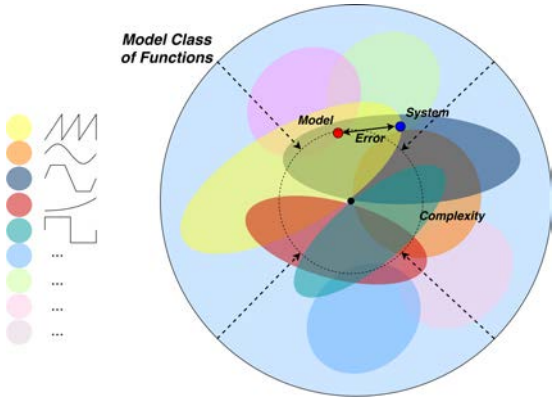



FIGURE 1.6 – In a *nonparametric* context, an *a priori* flexible model class is defined, with complexity constraints becoming an essential part of the modelling problem.

Here, $\beta \in \mathbb{N}_{\mathrm{o}}^{n_\theta}$ is a set of distinct, nonnegative integers with $\beta_i > \beta_{i-1}$. Each component $\phi(\cdot, \beta_i)$ of the model $\mathcal{M}_{nl,par}$ relates to a particular aspect of the model behaviour, and each weight $\alpha_i$ is a parameter of the model, indictating how important each component of the model is. Such a formulation can be described as *parametric*, as both the number of parameters and the values of those parameters are somehow informative in the model definition.

Whilst polynomials are very useful in certain types of problems, in practice care should be taken when trying to construct more complex functions as the presence of higher order terms ($\beta_i > 3$) quickly introduces unpredictable behaviour into the system. However, many other classes of nonlinear mappings exist and are also well-studied in the literature [Judistky et al., 1995, Nelles, 2001], e.g. local polynomial methods, orthogonal bases (OBFs) and wavelet expansions - to name but a few.

Evaluation of both the model class and the model complexity is now more difficult than the linear case (1.10). The model class is related to the mapping $\phi$, the number of components $n_\theta$, and the properties of each component $\beta$ - and the complexity depends on all of these variables plus the values of $\alpha$.

Intuitively, we can see that this is already less straightforward. Would a model with fewer components of greater order (e.g. $\beta = \{5, 6\}$) be considered less complex than a model with more components of lower order (e.g. $\beta = \{0, 1, 2, 3, 4\}$) ? Similarly, would a model composed of $n$ polynomial components be considered more or less complex than a model with $n$ harmonic components ? In both cases the answer is, rather unhelpfully, *'it depends'*. It could depend on the system, on the quality of the data obtained, on the properties of models themselves, or even on the envisaged application.

If good *a priori* knowledge of the system is available, this can be used to help the model selection process, in addition to model selection methods available in the literature (see for example, [Hong et al., 2008]). However, without such information the model class selection problem quickly becomes computationally challenging - requiring optimisation over many different aspects of the model definition.

**Nonparametric Methods**

This problem has led to an increasing interest in methods that allow the definition of an *a priori* flexible model class, such as *nonparametric methods* [Pillonetto et al., 2014]. In this sense, nonparametric does mean not that there are no parameters, but rather that the number of parameters is not fixed. Usually, the number of parameters is very large, and depends on the number of observations, e.g.

$$\mathcal{M}_{nl,npar}: \quad f(x_k) = \sum_{i=1}^{N} \alpha_i \, \phi(x_k, \gamma), \quad \alpha, \in \mathbb{R}^N, \gamma \in \mathbb{R}. \tag{1.13}$$

By using a large number of components, many different types of nonlinear functions can be reconstructed even through the definition of a single nonlinear mapping $\phi$ (as illustrated in Figure 1.6).

Hence, the model class definition is greatly simplified in this context: it depends solely on $\phi$ and $\gamma$ - where $\gamma$ represents some hyperparameter additionally configuring $\phi$ - and no longer on $n_\theta$ (which is N in (1.13)). The model parameters $\alpha$ no longer possess physical significance, in that examination of a particular value $\alpha_i$ tells us very little about the overall behaviour of the model.

However, the question of how to evaluate the model complexity still remains - and in fact, is now crucial. Clearly, it depends upon both $\phi$ and $\alpha$ - but the way in which it does so may depend on the true system or the intended purpose of the model. Furthermore, the consequence of using such a flexible model class definition is that constraints on the model complexity are essential to ensure the performance of the model.

In fact, this is what we will explore in this thesis: how different types of complexity constraints may be interesting in different scenarios, and how they can be incorporated into a framework consistent with the existing literature. The end of goal of this work is to move towards a more application-oriented framework for nonlinear identification, whereby the user is free to decide what characteristics are important in the estimation of nonlinear model, and subsequently tailor the estimation process to allow the development of a model best suited to their needs.

## 1.3 Thesis Overview

This thesis will focus exclusively on the application of kernel-based nonparametric methods to nonlinear identification problems.

As for other nonlinear methods, two key questions in kernel-based identification are the questions of *how to define a nonlinear model* (kernel selection) and *how to tune the complexity* of the model (regularisation). The following chapter will discuss how these questions are usually dealt with in the literature.

The principal contribution of this thesis is the presentation and investigation of two optimisation criteria (one existing in the literature and one novel proposition) for structural approximation and complexity tuning in kernel-based nonlinear system identification. Both methods are based on the idea of incorporating feature-based complexity constraints into the optimisation criterion, by penalising derivatives of functions. Essentially, such methods offer the user flexibility in the

definition of a kernel function and the choice of regularisation term, which opens new possibilities with respect to how nonlinear models can be estimated in practice.

Both methods bear strong links with other methods from the literature, which will be examined in detail in Chapters 2 and 3 and will form the basis of the subsequent developments of the thesis. Whilst analogy will be made with parallel frameworks, the discussion will be rooted in the framework of *Reproducing Kernel Hilbert Spaces* (RKHS). Using RKHS methods will allow analysis of the methods presented from both a theoretical and a practical point-of-view.

Furthermore, the methods developed will be applied to several identification 'case studies', comprising of both simulation and real-data examples, notably:

- Structural detection in static nonlinear systems.
- Controlling smoothness in LPV models.
- Complexity tuning using structural penalties in NARX systems.
- Internet traffic modelling using kernel methods.

### 1.3.1 Structure of the Thesis

The thesis will be structured as follows :

- *Chapter 2: Reproducing Kernels in Nonlinear System Identification*

  The following chapter will introduce the current state-of-the-art with respect to kernel methods in nonlinear system identification, with emphasis on two approaches: RKHS and *Sobolev space* methods. The discussion will centre around the question of choosing a kernel function, and in particular, how this relates to the model class definition in a classical identification setting.

- *Chapter 3: Penalising Derivatives in the RKHS*

  In chapter three, the discussion will turn to the developments of this thesis, i.e. the formulation of methods for penalising derivatives in an RKHS. Two methods will be presented, one from the literature and one novel proposition. Each optimisation scheme will be formulated carefully, with attention given to both theoretical and practical considerations. To conclude, a comparative example will be presented using all the methods formulated up to this point. The aim of this example will be to illustrate the theory discussed, and provide an objective comparison between the performance of the methods presented in chapters two and three.

- *Chapter 4: Identification Case Studies*

  In chapter four, the methods presented in the previous chapter will be applied to several simulation examples, with the aim of showing how they can be used in practice. Three distinct scenarios will be considered, with each example selected to illustrate a different aspect of the methods presented:
  - Structural detection in static nonlinear systems.
    - It will be shown that using a derivative penalty offers a way to deconstrain the model class definition in small problems, offering improved accuracy and physical insight in certain types of problems.
  - Controlling smoothness in LPV models.
    - To illustrate how derivative penalties can be applied to the types of dynamical model structures of interest in the control and identification communities, the methods

presented will be formulated for the case of LPV models.

· Complexity tuning using structural penalties in NARX systems.
  — It will also be shown how derivative penalties can be used to formulate structural constraints, offering an entirely new way of acting on the properties of a model. To the best of the author's knowledge, there is little or no discussion of such methods in the literature, as achieving similar control over the model properties in other kernel-based frameworks is extremely difficult.
  — Structural constraints could be of interest in many ways, e.g. structural detection or approximation and variable selection. Here we will consider just one possible application, namely how structural constraints can be used in addition to existing techniques, offering an additional degree of freedom in the optimisation problem and consequently providing a simple way of improving performance in nonlinear modelling.

- *Chapter 5: Application to Real Data*

  The final part of this thesis will focus on an application to a real-data problem: forecasting network traffic for network diagnostics and maintenance.
  This chapter documents the first part of an ongoing collaboration with an industrial partner, and will discuss the need for models and the challenges involved in modelling network data.
  It will be shown that a kernel-based approach provides an accurate, flexible and user-friendly approach for such problems, and how derivative regularisation can be used in this context to simplify the modelling problem.

### 1.3.2 Associated Publications

1. *The Impact of Smoothness on Model Class Selection in Nonlinear System Identification: An Application of Derivatives in the RKHS*
   Y. Bhujwalla, V. Laurain and M. Gilson
   Presented at the 2016 American Control Conference (ACC), Boston, USA.
2. *An RKHS Approach to Systematic Kernel Selection in Nonlinear System Identification*
   Y. Bhujwalla, V. Laurain and M. Gilson
   Presented at the 2016 IEEE Conference on Decision and Control (CDC), Las Vegas, USA.
3. *An AS-Level Approach to Network Traffic Analysis and Modelling*
   Q. Grandemange, Y. Bhujwalla, M. Gilson, O. Ferveur and E. Gnaedinger
   Presented at the 2017 IEEE International Conference on Communications (ICC), Paris, France.
4. *How We Spend Our Time Online: Predicting Network Traffic Using System Identification*
   Y. Bhujwalla, Q. Grandemange, M. Gilson, V. Laurain and E. Gnaedinger
   Presented at the 2017 IFAC World Congress, Toulouse, France.
5. *An RKHS Approach to Controlling Smoothness in Nonparametric LPV-IO Identification*
   Y. Bhujwalla, V. Laurain and M. Gilson
   Presented at the 2017 IFAC World Congress, Toulouse, France.

# 2

# Reproducing Kernels in Nonlinear System Identification

*In this chapter, a review of current approaches to kernel-based system identification will be presented, with particular emphasis on two approaches (RKHS and Sobolev Space methods). A theoretical framework for the problem will be formulated, but the discussion will focus on practical concerns and user choices - in particular, the question of how to choose a kernel function.*

## 2.1 Introduction

One approach to nonlinear identification problems that has garnered increasing attention in recent years is that of a kernel-based approach. Whilst kernel methods are relatively recent addition to the canon of identification methods, their adoption in other statisctical communities is far more widespread, dating back to the mid-twentieth century [Aronszajn, 1950, Krige, 1966, Parzen, 1960], and in some cases even earlier e.g. [Mercer, 1909, Moore, 1916].

Kernel methods provide a flexible and mathematically-elegant framework for dealing with a range of estimation problems: by relating them to a functional reconstruction problem in a higher-dimensional feature space. By controlling the features of this space, the properties of models - or functions - are also implicitly controlled. Furthermore, they provide a consistent framework for dealing with many different types of problems, which in many cases greatly reduces the effort required to derive solutions.

Whilst well-known in the statistical and time-series communities, arguably it wasn't until the development of *support vector machine* classifiers (SVMs) [Cortes and Vapnik, 1995], and their regression counterpart *least-squares support vector machines* (LS-SVMs) [Suykens et al., 2002], that such methods were popularised in the machine learning community and elsewhere. In an SVM approach, problems are formulated in a *primal* (input) space, whilst they can be solved in either the primal space or in a *dual* (nonparametric feature) space as required - allowing for easier handling of high-dimensional problems and large datasets.

For SVM classifiers, the solution defines a decision boundary - separating the data in the 'best' possible manner. In the LS-SVM formulation, the decision boundary becomes a predictive model,

by using a *least-squares* rather than $\epsilon$*-sensitive* loss-function. In both cases, the kernel function acts as a conduit between the primal and dual spaces, allowing high (and even infinite) dimensional feature spaces to be accessed indirectly.

Kernel methods can also be extended from a deterministic to a stochastic framework, via Bayesian or robust statistical methods, either through LS-SVM methods (see e.g. [De Brabanter et al., 2011]) or through a *Gaussian Process* approach.

*Gaussian Process* (GP) regression relates the kernel function to the covariance function of a stochastic random process [MacKay, 1998, Rasmussen and Williams, 2006]. Along with LS-SVMs, GPs are arguably the dominant paradigm in the identification literature to date. And importantly, there is a known equivalence between the two methods: for equivalent configurations, the estimated model in the LS-SVM case is the mean of the corresponding Gaussian process. In practice, the estimated models can differ quite signficantly, as a result of practical consequences (e.g. model selection criteria) associated with each framework.

Although their popularity in system identification is relatively recent, both methods are arguably well-suited to identification problems, and as such have already been successfully applied to a range of different problems, for example:

- LTI identification [Pillonetto and De Nicolao, 2010, Chen et al., 2012b],
- Frequency-Domain identification [Lataire and Chen, 2016, Marconato et al., 2016],
- LPV identification [Tóth et al., 2011, Tóth et al., 2011, Piga and Tóth, 2013],
- Wiener-Hammerstein identification [Goethals et al., 2005b, Marconato and Schoukens, 2009, Risuleo et al., 2015a],
- Estimation under generalised noise conditions [Laurain et al., 2012, Darwish et al., 2015, Laurain et al., 2015],
- Nonlinear control [Suykens et al., 2001, Hall et al., 2012],

and many others besides.

And interestingly, both paradigms can be linked to another increasingly well-studied framework: *Reproducing Kernel Hilbert Space* (RKHS) methods [Aronszajn, 1950]. RKHS methods allow us to make a conceptual link between popular kernel-based methodologies and classical identification notions, such as the model class, model structure and complexity. In a similar manner to an SVM approach, the kernel acts as a mapping between the input space and a feature space, however in this case the problem is formulated directly in an RKHS.

This chapter aims to clearly illustrate how nonlinear models can be estimated in an RKHS, with a focus on the relation between theoretical notions, such as functional spaces and uniqueness of functions, and practical choices such as the definition of a model, model class and model structure. Sections 2.2 of this chapter will show how functions can be constructed in an RKHS, and how their properties can be controlled in practice, whilst section 2.3 will extend the theory to identification problems.

In addition, we will consider the estimation of functions in a *Sobolev Space* [Adams, 1975, Wahba, 1990]. As will be shown, Sobolev methods are closely related to RKHS methods. However, in addition to achieving control over the function through the kernel, properties are also encoded into the definition of the functional norm.

This offers an altogether different way of acting on the model properties. Not only can we act on the mapping from the data to the feature space (through the kernel), but also on the mapping

from the feature space to the model (through the model parameters). And as will be shown in section 2.4, the two can be used to achieve very similar results.

## 2.2 Reproducing Kernels

In this section, several key results regarding *Reproducing Kernel Hilbert Spaces* and *Sobolev Spaces* will be presented. The interest, from the author's perspective, in considering RKHS and Sobolev space methods is that both provide a solid theoretical framework that allows the interpretation of and generalisation to other kernel-based formulations.

Furthermore, they provide a highly conceptual framework in which results can be both developed and understood. As engineers, this is of course very appealing. Whereas mathematicians deal with axioms, and physicists with laws, we deal with assumptions. *'All models are wrong...'*, and understanding the implications of the assumptions we make is crucial in ensuring that our models are not any more wrong than they need to be.

### 2.2.1 Reproducing Kernels Hilbert Spaces

RKHS theory links the definition of a kernel function to the explicit characterisation of a functional space. Different functions are essentially defined as would be different vectors in a vector space, with the functions inheriting the properties of the space - as specified by the kernel.

More formally, an RKHS is a *Hilbert Space* $\mathcal{H}$, such that it is a complete metric space, in which functions can be related to each other and themselves through an inner-product and norm [Akhiezer and Glazman, 1963, Halmos, 1957]:

$$\textbf{Inner-Product:} \qquad \text{for } f, g \in \mathcal{H} \quad \exists\, \langle f, g \rangle_{\mathcal{H}}, \tag{2.1}$$

$$\textbf{Norm:} \qquad \|f\|_{\mathcal{H}} = \sqrt{\langle f, f \rangle_{\mathcal{H}}} \geq 0. \tag{2.2}$$

Whilst all RKHS are Hilbert spaces, not every Hilbert space is an RKHS. For example, the space of square integrable $\mathcal{L}_2$ functions is a Hilbert space but not an RKHS [Rasmussen and Williams, 2006]:

$$\|f\|_2 = \Big( \int_{\mathcal{X}} f(\mathbf{x})^2 \, \mathrm{d}x \Big)^{1/2}. \tag{2.3}$$

In fact, for a Hilbert space to be an RKHS it must be possible to define a linear functional from $\mathcal{H}$ to $\mathbb{R}$, $\mathcal{L}_x : f \to f(x)$, such that $\mathcal{L}_x$ is continuous over all $f \in \mathcal{H}$ [Kimeldorf and Wahba, 1971]. Or in simpler terms, an RKHS is a Hilbert space in which pointwise evaluation of $f$ is a continuous linear functional.

#### The Reproducing Property

More formally, this statement relates to the *Riesz Representation Theorem* [Riesz and Szőkefalvi-Nagy, 1953], and its definition implies the definition of the first of two fundamental results in RKHS theory: *The Reproducing Property*. To simplify the presentation, the formulation will be restricted to Hilbert spaces of real-valued functions.

From [Aronszajn, 1950], the reproducing property states that if a function $f$ lies in an RKHS $\mathcal{H}$, it is possible to evaluate $f$ at points in $\mathbb{R}$ using linear combinations of *kernel slices* $k_{\mathbf{x}} : \mathcal{X} \to \mathbb{R}$:

$$f(\mathbf{x}) = \langle f, k_{\mathbf{x}} \rangle_{\mathcal{H}}. \tag{2.4}$$

**The Kernel Trick:**

As $k_{\mathbf{x}}$ is also a function in $\mathcal{H}$, it can be seen that $k_{\mathbf{x}}$ is a *reproducing kernel*, in that:

$$\begin{aligned} \langle k_{\mathbf{x}_i}, k_{\mathbf{x}_j} \rangle_{\mathcal{H}} &= k_{\mathbf{x}_i}(\mathbf{x}_j) \\ &= k_{\mathbf{x}_j}(\mathbf{x}_i) \\ &= \mathrm{K}(\mathbf{x}_i, \mathbf{x}_j), \end{aligned} \tag{2.5}$$

where $\mathrm{K} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a *kernel function*. This result is also known as *the kernel trick*, and is the second our two fundamental results in RKHS theory. Its definition implies several important points about the function K:

1. An RKHS kernel function must be symmetric, as $k_{\mathbf{x}_i}(\mathbf{x}_j) = k_{\mathbf{x}_j}(\mathbf{x}_i)$.
2. An RKHS kernel function must be positive semi-definite, as $\mathrm{K}(\mathbf{x}_i, \mathbf{x}_i) = \langle k_{\mathbf{x}_i}, k_{\mathbf{x}_i} \rangle_{\mathcal{H}} \geq 0$.

Furthermore, it illustrates one of the most appealing parts of RKHS theory. Through the kernel trick, the computation of dot-products and integrals is reduced to the much simpler evaluation of kernel functions. In this way, irrespective of the complexity of $\mathcal{H}$ ($\mathcal{H}$ can even be an infinite dimensional space), the difficulty of evaluating expressions does not change. Hence, very complex problems can be embedded into the same framework as very simple problems - with no increase in computational effort.

**The Moore-Aronszajn Theorem:**

The above results can be restated in the opposite sense. From [Mercer, 1909, Moore, 1916, Aronszajn, 1950], *The Moore-Aronszajn Theorem* states that any symmetric, positive definite function is a Mercer kernel $\mathrm{K} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ defining a unique, corresponding RKHS $\mathcal{H}$, such that for functions $f(\mathbf{x}) = \sum_i \alpha_i k_{\mathbf{x}_i}(\mathbf{x}) \in \mathcal{H}$ and $g(\mathbf{x}) = \sum_j \beta_j k_{\mathbf{x}_j}(\mathbf{x}) \in \mathcal{H}$:

$$1. \qquad \langle f, g \rangle_{\mathcal{H}} = \sum_i \sum_j \alpha_i \beta_j \mathrm{K}(\mathbf{x}_i, \mathbf{x}_j), \quad \alpha, \beta \in \mathbb{R}^{\infty}. \tag{2.6}$$

$$\begin{aligned} 2. \qquad \|f\|^2_{\mathcal{H}} &= \langle f, f \rangle_{\mathcal{H}} \\ &= \langle \sum_i \alpha_i k_{\mathbf{x}_i}, \sum_j \alpha_j k_{\mathbf{x}_j} \rangle_{\mathcal{H}} \\ &= \sum_i \sum_j \alpha_i \alpha_j \mathrm{K}(\mathbf{x}_i, \mathbf{x}_j) \\ &= \alpha^{\top} \mathbf{K} \alpha, \end{aligned} \tag{2.7}$$

where $\mathbf{K}$ is the *Gram Matrix* such that $\{\mathbf{K}\}_{i,j} = \mathrm{K}(\mathbf{x}_i, \mathbf{x}_j)$ and its positive semi-definiteness implies that $\mathbf{c}^{\top} \mathbf{K} \mathbf{c} \geq 0$, for any $\mathbf{c} \in \mathbb{R}^{\infty}$.

### 2.2.2 Choosing a Kernel Function: Hard Bounds

As $\mathbf{K}$ defines a mapping from the input space (corresponding to the input data) to the feature space $\mathcal{H}$, the properties of functions in $\mathcal{H}$ are strongly linked to the properties of

the mapping $\mathbf{K}$. By varying the choice of kernel function, many different types functions can be defined. Clearly then, proper selection of the kernel matrix is a crucial stage in the identification process and much attention has been dedicated to it in the literature, including mathematical perspectives [Zhou, 2002, Steinwart and Christmann, 2008], machine learning perspectives [Rasmussen and Williams, 2006, Duvenaud, 2014] and identification perspectives [Pillonetto et al., 2014].

Functions in the RKHS are defined by choosing a mapping from $\mathcal{X}$ to $\mathcal{H}$, and a set of weights specifying a particular function in chosen feature space. As any combination of weights $\alpha_i \in \mathbb{R}$ defines a valid RKHS function, it is only the properties of $\mathcal{H}$ that limit the types of functions that can be represented.

Therefore, we can interpret the feature space $\mathcal{H}$ as being equivalent to the notion of a model class $\mathcal{M}$ in an identification sense. And as $\mathcal{H}$ is fully specified through $\mathbf{K}$, the problem of defining a model class (discussed in section 1.2.3) is equivalent to the selection of a kernel function. And, just as the model class places strict limits on the types of models that can be constructed, a kernel places strict limits on the properties of $f$ - a function cannot be outside the model class (as depicted in Figure 2.1). In many cases, such limits are desirable - for example by allowing us to restrict $\mathcal{H}$ to include only functions of particular interest, such as Wiener-Hammerstein, LPV or stable FIR models.

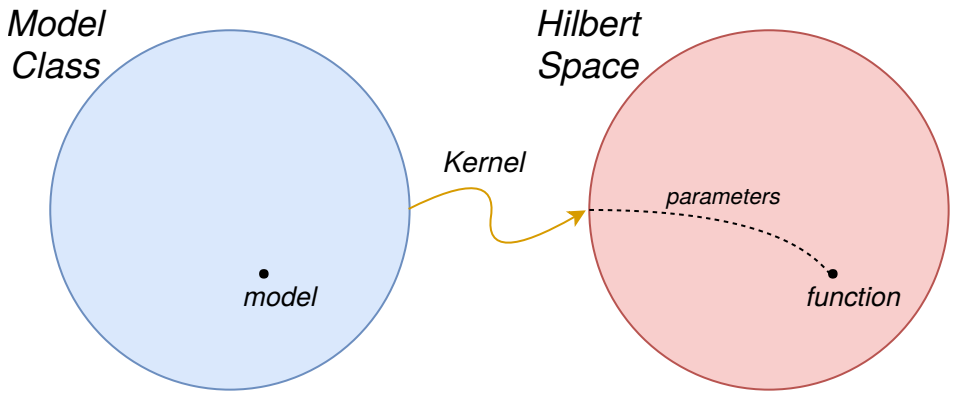



FIGURE 2.1 – Model class selection in the RKHS : $\mathcal{S} \in \mathcal{M} \rightarrow f \in \mathcal{H}$

In this section, we will give some examples of how the choice of kernel can be controlled. All will be types of kernels that place clear limits on the type of functions that can subsequently be represented. These types of limits will be referred to as *hard bounds*, in the sense that not only are the properties of the model class strictly limited by the choice of kernel - but also all functions in the model class exhibit broadly similar properties in at least one respect.

Principally, there are three ways in which the choice of kernel can be controlled, which will be referred to here as *shape, scale* and *structure*:

1. **Shape** $\longrightarrow$ the type of kernel function used.
2. **Scale** $\longrightarrow$ the tuning of any kernel hyperparameters.
3. **Structure** $\longrightarrow$ the combination of and interactions between input variables.

**Shaping the Kernel**

As stated in section 2.2, any symmetric positive-definite function is a valid kernel, defining a corresponding RKHS. Many kernels exist, and can be categorised in different way depending on the perspectives of the user and their chosen application. Here we will give just some examples of kernel functions, as this topic is well-addressed in the literature.

*1. The Linear Kernel*

The simplest example of this is the one-dimensional linear kernel :

$$\mathrm{K}(x_i, x_j) = x_i x_j, \quad x_i, x_j \in \mathbb{R} \tag{2.8}$$

which defines a space of linear functions $f(x) = cx$ for some constant $c \in \mathbb{R}$. It might seem a little strange to go through the development of an entire theory of functional spaces and apply it to the estimation of linear functions - a relatively well-understood topic - but in fact, this kernel is widely-used in many practical applications such as text classification and natural language processing [Schölkopf and Smola, 2002] and shows that kernel methods can include and generalise linear methods.

*2. The Polynomial Kernel*

Sums and products of kernels are also kernels [Aronszajn, 1950], meaning that combinations of linear kernels can be used to form polynomial kernel functions :

$$\mathrm{K}(x_i, x_j) = (x_i x_j + \gamma_1)^{\gamma_2}, \quad x_i, x_j, \gamma_1 \in \mathbb{R},\ \gamma_2 \in \mathbb{N} \tag{2.9}$$

where $\gamma_1$, $\gamma_2$ are *hyperparameters* additionally configuring the properties of $\mathbf{K}$. Alternatively, we can view the linear kernel as a special case of (2.9) for $(\gamma_1, \gamma_2) = (0, 1)$.

*3. Radial Basis Function Kernels*

Both (2.8) and (2.9) are examples of *non-stationary* kernels, i.e. kernels which depend on the absolute location of the inputs. *Stationary* kernels, i.e. kernels that are *translation invariant* (depending only on the relative position of their inputs and not their absolute position), are also very popular. Such kernels act as similarity measures, whereby the proximity of locations in the input space determines how much observations at given points influence each other. For example, consider the class of *radial basis functions* (RBFs) :

$$\mathrm{K}(x_i, x_j) = g\left(\gamma_1, \|\mathbf{x}_j - \mathbf{x}_i\|^{\gamma_2}\right). \tag{2.10}$$

The choice of $g(\cdot)$ and $\gamma_2$ control the profile of the kernel, with $\gamma_1$ usually acting as a some sort of scaling or offset factor. Different choices of $g(\cdot)$ can be found in the literature, with best known example being the class of exponential radial basis functions $g\left(\gamma_1, \|\mathbf{x}_j - \mathbf{x}_i\|^{\gamma_2}\right) = \exp\left(-\gamma_1 \|\mathbf{x}_j - \mathbf{x}_i\|^{\gamma_2}\right)$ for $\gamma_1 > 0$. In this case, the change to the behaviour of the kernel induced by different values of $\gamma_2$ is so pronounced that it is usually used to designate different kernels.

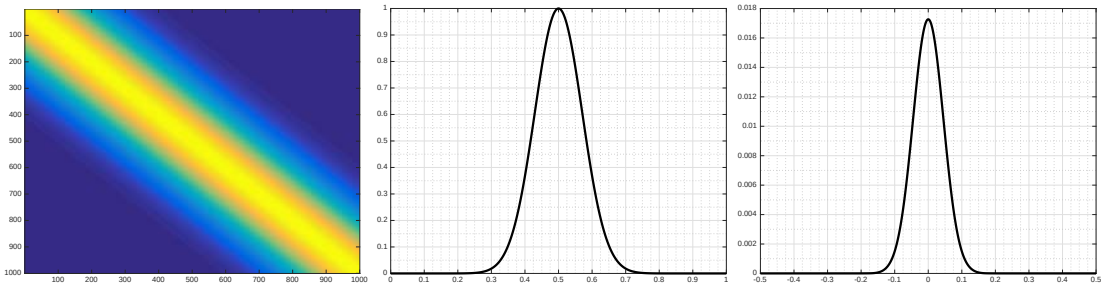
*4. The Gaussian RBF Kernel*

For example, letting $\gamma_2 = 1$ gives a nonsmooth exponential kernel (sometimes referred to as the Laplacian kernel), which has a very sharp roll-off away from its centre. Alternatively, letting $\gamma_2 = 2$ gives the Gaussian RBF kernel :

$$\mathrm{K}(x_i, x_j) = \exp\left\{-\frac{\|x_j - x_i\|^2}{\sigma^2}\right\}, \tag{2.11}$$

where $\gamma_1$ is replaced with $\sigma^2$ to denote its relation to the standard deviation of the Gaussian distribution.

The Gaussian kernel is probably the most widely used kernel function in practice, and as such its properties have been extensively studied in the literature (see for example [Steinwart et al., 2006, Minh, 2010]). Not only is it capable of reproducing a wide range of smooth nonlinear functions, but it has many convenient mathematical properties that make it very easy to work with [Poggio and Girosi, 1990]. Futhermore, it has even been investigated in a *stable* form, using a multiplicative decay factor to allow the estimation of stable impulse response models [Pillonetto et al., 2011, Darwish et al., 2015].



(a) $\mathbf{K}$ (b) $\mathrm{K}(\cdot, \mathbf{x})$ (c) $\mathcal{F}\{\mathrm{K}(\cdot, \mathbf{x})\}$

FIGURE 2.2 – The Gaussian radial basis function kernel (2.11).

Figure 2.2 illustrates the Gaussian kernel evaluated over a grid of values $\mathbf{x} \in [0, 1]^{1000}$, with Figure 2.2a depicting the kernel as it would appear in (2.31), Figure 2.2b showing a diagonal cross-section of the kernel and Figure 2.2c showing the Fourier transform of such a cross-section.

As can be seen in Figure 2.2c, the Gaussian kernel is unique in that its spectra is also a Gaussian. Hence, it acts in a similar manner to a low-pass smoothing of the observations, with $\sigma$ controlling the bandwidth of the filter.

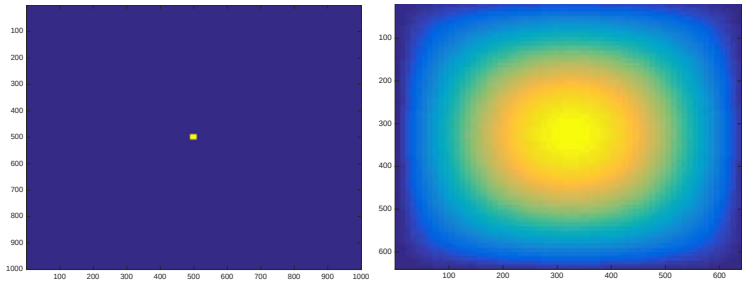
### Scaling the Kernel

It is tempting to think of the type of kernel function as the key component in the specification of $\mathcal{H}$, with any associated hyperparameters merely acting as some sort of tuning factor. But this is misleading: the hyperparameters also impact upon the properties of $\mathcal{H}$, albeit in a slightly different manner.

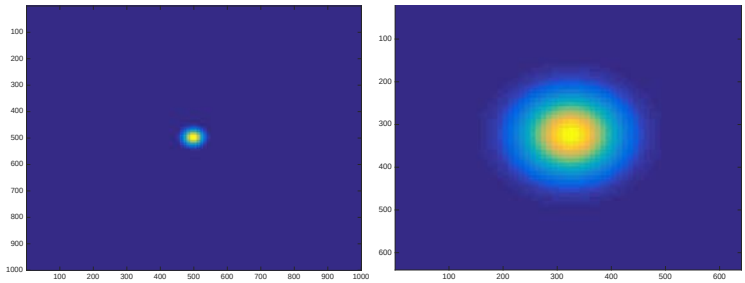
This confusion stems principally from two factors. Firstly, hyperparameters are often straightforward to include in the optimisation problem, and can be determined using a suitable selection criterion (such as marginal likelihood or cross-validation). Secondly, they offer a way to *continuously* tune the features of the model - in a similar manner to the tuning of a regularisation hyperparameter.

But these two things should not be confused. Whilst kernel hyperparameters do offer a continuous manner of tuning the model properties, they still place *hard bounds* on the model class - in the same fashion as the choice of kernel function.

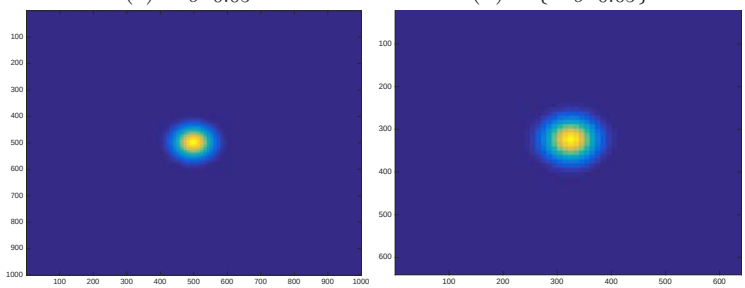
As an example, consider the Gaussian RBF kernel in (2.11). We can consider $\sigma$ as being related to the domain of influence of individual points in $\mathcal{X}$.



(a) $\mathbf{K}_{\sigma=0.01}$ (b) $\mathcal{F}\{\mathbf{K}_{\sigma=0.01}\}$



(c) $\mathbf{K}_{\sigma=0.05}$ (d) $\mathcal{F}\{\mathbf{K}_{\sigma=0.05}\}$



(e) $\mathbf{K}_{\sigma=0.1}$ (f) $\mathcal{F}\{\mathbf{K}_{\sigma=0.1}\}$

FIGURE 2.3 – The width of a Gaussian RBF in the spatial domain (left) controls its band in the spatial frequency domain (right).

- $\sigma \to 0$: Intuitively, we can see that as $\sigma$ approaches zero, K tends to something increasingly resembling a dirac function $\delta(x_j - x_i)$ - where points in $\mathcal{X}$ are completely independent of each other, and no inference is made anywhere.
- $\sigma \to \infty$: At the other extreme, as $\sigma$ tends to infinity, K tends to 1. In this case, all points influence each other equally - and the resulting model would simply be the average of all observations.

Alternatively, we can think of $\sigma$ as placing a hard *lower bound* on the smoothness of the model, and an *upper bound* on its flexibility. This is illustrated in Figure (2.3), where a zero-centered Gaussian kernel is evaluated over a two-dimensional grid between -1 and 1 for different values of $\sigma$. The left-hand images show the kernel evaluations, whilst the right-hand images show the 2D spectra of these images. Note that $\mathbf{K}$ here is formed by taking the element-wise products of the one-dimensional Gaussian kernel,

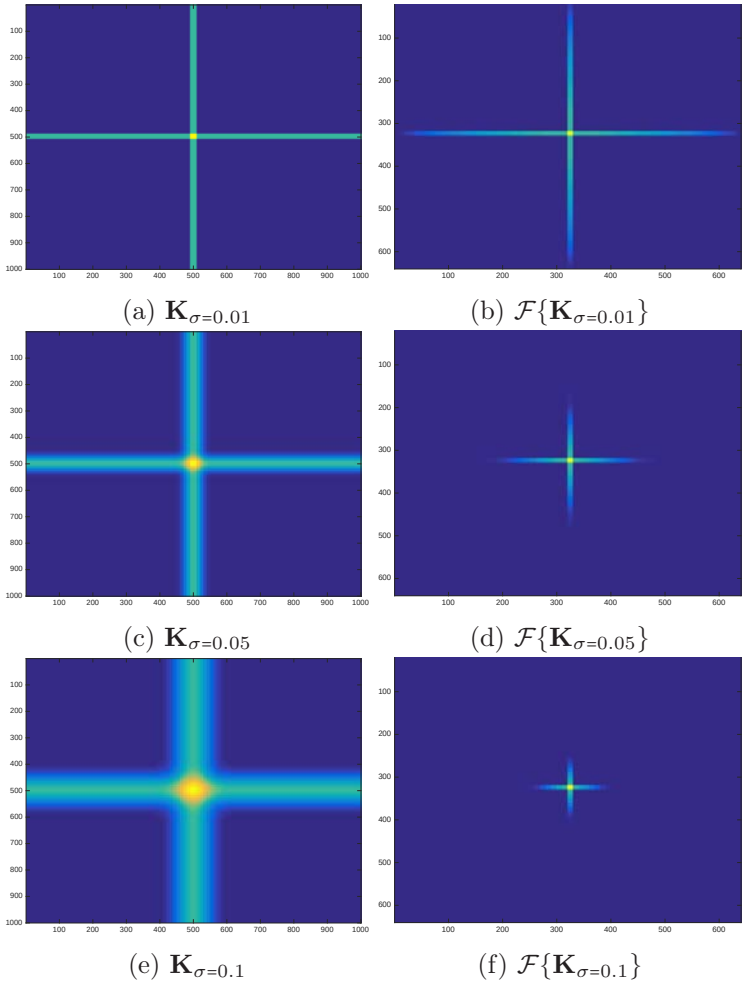
$$\mathrm{K}(\mathbf{x}_i, \mathbf{x}_j) = \mathrm{K}(x_{1,i}, x_{1,j}) \times \mathrm{K}(x_{2,i}, x_{2,j}). \tag{2.12}$$

As can be seen, a small kernel corresponds to a flexible model class: i.e. a large band in the frequency domain. As the kernel broadens, high-frequency components are progressively attenuated,

resulting in a smoother, less flexible model.

### Structure in Kernels

The final part of this discussion on kernel methods is focused on the link between interactions of variables and the model class: and how this can be controlled through the kernel function.



(a) $\mathbf{K}_{\sigma=0.01}$ (b) $\mathcal{F}\{\mathbf{K}_{\sigma=0.01}\}$

(c) $\mathbf{K}_{\sigma=0.05}$ (d) $\mathcal{F}\{\mathbf{K}_{\sigma=0.05}\}$

(e) $\mathbf{K}_{\sigma=0.1}$ (f) $\mathcal{F}\{\mathbf{K}_{\sigma=0.1}\}$

FIGURE 2.4 – An additive Gaussian kernel has the same low-pass filtering effect as a multiplicative Gaussian kernel, however interactions between variables are filtered out.

One of the most appealing features of an RKHS approach to identification problems is that, through different combinations (i.e. sums and products) of different kernels, the same framework can be readily applied to different nonlinear problems. In system identification, this characteristic has been repeatedly exploited to allow flexible modelling techniques to be used in control-relevant structures, such as Wiener-Hammerstein and LPV models.

However, less explored in the identification literature is the usage of structural constraints as a way of tuning the complexity of the model, by reducing the effective size of the model class. For example, forming a multidimensional Gaussian kernel as a *sum* of kernels - rather than a product - allows for nonlinearities with respect to individual variables, whilst neglecting the effect

of interactions between variables in the model.

$$\mathrm{K}(\mathbf{x}_i, \mathbf{x}_j) = \mathrm{K}(x_{1,i}, x_{1,j}) + \mathrm{K}(x_{2,i}, x_{2,j}). \tag{2.13}$$

Such a kernel has the same smoothing properties as the usual Gaussian kernel, but with a greatly reduced band of passable frequencies. This is illustrated in Figure 2.4, which shows the smoothing characteristics of an additive Gaussian kernel in the similar manner to Figure 2.3.

Hence, imposing structure on the kernel not only allows the estimation of control-suitable models but also provides an effective way of acting on the model complexity: overcoming the curse of dimensionality and allowing efficient low-variance estimates for high-dimensional problems [Duvenaud et al., 2011, Duvenaud et al., 2013].

In this setting, structural choices can only be made in a *discrete fashion* (e.g. to permit or not to permit variable interactions in the model), unlike the continuous tuning offered by the kernel hyperparameters. Whilst this does open many possibilities, such constraints may be overly restrictive, particularly if their purpose is to improve the numerical properties of the model, rather than to enforce a structure *a priori* known to be similar to the true system (akin to a grey-box modelling approach).

### 2.2.3 Choosing a Kernel Function: Soft Bounds

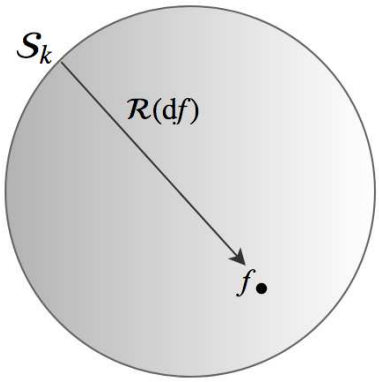



FIGURE 2.5 – In a Sobolev space, the kernel function maps to an *a priori* flexible model class, with different functions within the feature space exhibiting different properties.

In the previous section, examples of kernel functions that place what was referred to as hard bounds were presented. In this situation, the kernel places strict limits on the types of functions that can be represented, and functions in a space will share similar properties. For example, the Gaussian RBF kernel defines a space of smooth functions, with the kernel width controlling the frequency band of the feature space. Whilst many different functions can be represented in this formulation, all will be smooth functions - with the smoothness controlled by the kernel hyperparameter.

However, certain classes of kernel functions have a particular structure through which they define classes of functions with greatly different properties. In this scenario, the model properties are not only determined by the kernel mapping by also heavily determined by the values of the weights

$\alpha$ (as illustrated in Figure 2.5). In fact, as will be discussed, whilst such kernels stil define an RKHS, this RKHS is additionally constrained in the way functions are evaluated in the space - and corresponds to a *Sobolev Space* [Adams, 1975].

In this section a framework for *soft* model class selection will be formulated using methods form the literature. It will be shown in section 2.4 that for certain problems, this can allow for a level of control over the model properties equivalent to the techniques presented in section 2.2.

The term *soft* constraints is used here to refer to constraints placed directly on the overall properties of the model - rather than strict bounds on the functional space. This can be though of as a *top-down* constraint, in which the model class is implicitly constrained by the demands placed on the model itself - as opposed to a *ground-up* approach, where limits on the model class dictate the properties of the model.

The most well-known example of kernels defining a Sobolev space are the smoothing spline kernels, which are widely studied in both the statistical literature [Wahba, 1990, Berlinet and Thomas-Agnan, 2004,Ramsay and Silverman, 1997] and increasingly so also in system identification e.g. [Lataire and Chen, 2016, Pillonetto et al., 2011].

### Smoothing Splines

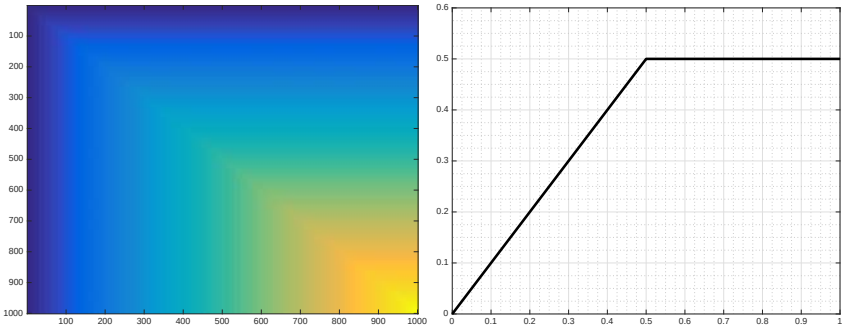
Smoothing splines are a well-known class of functions in statistical learning. They are inherently flexible, and can be used to represent a wide range of functions. Usually, the term smoothing splines is used to refer to functions defined over $\mathcal{X} \in \mathbb{R}$, however they are closely related to other types of spline functions that model multidimensional functions (e.g. thin-plate splines).

Essentially, a smoothing spline is a local polynomial kernel of a particular order, additional constrained such that functions are continuous up to a certain order at the locations of the kernels (often referred to in the literature as *knots* [Wahba, 1990]).

For example, the first-order smoothing spline is the *linear spline kernel*, depicted in Figure 2.6 [Pillonetto et al., 2014] :

$$\mathrm{K}(x_i, x_j) = \min(x_i, x_j). \tag{2.14}$$

In this case, the kernel function is a linear interpolation between points in the input space, which




(a) $\mathbf{K}$ (b) $\mathrm{K}(0.5, \mathbf{x})$

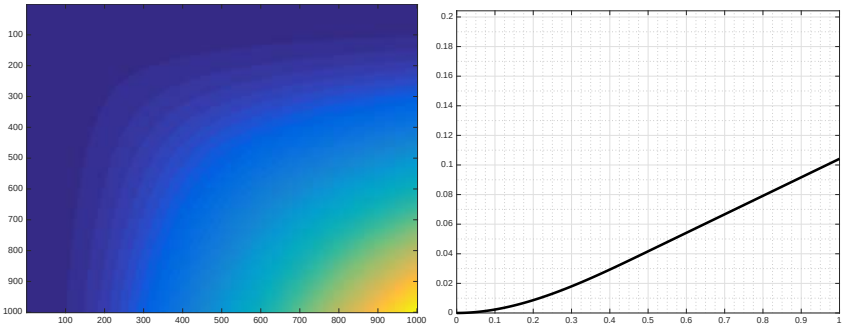
FIGURE 2.6 – The linear spline kernel (2.14).

is point-wise continuous at the knots.

Similarly, the *cubic spline kernel*, depicted in Figure 2.7, is a cubic local polynomial function whose first derivatives are continuous at the locations of the kernel functions [Pillonetto et al., 2014] :

$$\mathrm{K}(x_i, x_j) = \frac{x_i x_j \min(x_i, x_j)}{2} - \frac{\min(x_i, x_j)^3}{6}. \tag{2.15}$$

The latter is widely widely used in inverse problems as it provides visually smooth results and is capable of reconstructing a wide range of functions - whilst also being sufficiently rigid to avoid unpredictable behaviour.



(a) $\mathbf{K}$ (b) $\mathrm{K}(0.5, \mathbf{x})$

FIGURE 2.7 – The cubic spline kernel (2.15).

As can be seen in (2.14) and (2.15), unlike the Gaussian RBF, the smoothing spline functions above have no hyperparameters - instead defining an *a priori* flexible structure.

### Sobolev Spaces and Reproducing Kernels

As smoothing splines are already well-known in the literature, it may not seem necessary to consider any additional formulation in order to use them, but in fact they are an excellent example of how RKHS theory can help us to better understand the tools at our disposal.

In fact, the smoothing spline kernel relates to a specific type of functional space called a *Sobolev Space* $\mathcal{S}_m^p$ [Adams, 1975], in which functions are evaluated in terms of their derivative energies, e.g. for $\mathcal{X} = \mathbb{R}$

$$\| f \|_{\mathcal{S}_m^p} = \Big( \sum_{i=0}^{m} \int_{\mathcal{X}} \left( \frac{\mathrm{d}^i f(x)}{\mathrm{d}x^i} \right)^p \mathrm{d}x \Big)^{1/p}. \tag{2.16}$$

If $p = 2$, it can be seen that $\mathcal{S}_m^2$ is also an RKHS, with some additional constraints. As in an RKHS, $\mathcal{S}_m$ is spanned by an associated reproducing kernel function. Unlike in an RKHS, the choice of this function is not free - but rather implicit in the definition of the functional norm.

Therefore, $\mathbf{K}$ must be determined, either by formulating and solving the corresponding *Green's Function* problem - as discussed in [Wahba, 1990], or alternatively by analysis using Fourier methods - as illustrated in [Poggio and Girosi, 1990, Rasmussen and Williams, 2006].

In the case of the smoothing splines presented in the previous section, the Green's function is defined as:

$$\mathcal{G}_m(x, t) = \frac{(x-t)_+^{m-1}}{(m-1)!}, \qquad (z)_+ = \begin{cases} z, & z \geq 0 \\ 0, & z < 0, \end{cases} \tag{2.17}$$

with $\mathbf{K}$ found by solving:

$$\mathrm{K}(x_i, x_j) = \int_0^1 \mathcal{G}_m(x_i, t)\mathcal{G}_m(x_j, t)\mathrm{d}t. \tag{2.18}$$

The linear spline kernel and cubic spline kernel of the previous section correspond to the spaces $\mathcal{S}_1^2$ and $\mathcal{S}_2^2$ (i.e. spaces with norms evaluating the first and second derivatives respectively).

What is interesting is that we can see that functions estimated in $\mathcal{S}_m^2$ will have quite different properties: a function with a larger norm will also be less smooth - in the sense in which $\mathcal{S}_m^2$ is defined. Hence, a Sobolev kernel not only defines a space of smooth functions, but the smoothness of these functions varies throughout the space.

#### Extension to Multivariate Problems

Whilst smoothing splines provide a good illustrative example, and are also useful in practice, they are by no means the only interesting type of Sobolev kernel. Many examples of spline kernels with interesting properties can be found in the literature, such as periodic and partial splines [Wahba, 1990, Berlinet and Thomas-Agnan, 2004], and even splines applied to dynamical problems [Ramsay and Silverman, 1997, Ramsay and Silverman, 2002].

However, due to their mathematically-involved definition, their extension to multivariate problems is slightly less straightforward. Whilst Sobolev kernels are also RKHS kernels, and can therefore be combined as sums and products as discussed in section 2.2.2, this does not then automatically correspond to the definition of a multivariate Sobolev space. As in the smoothing spline case, determination of the proper kernel requires formulation and solution of the Green's function problem - which in practice may be non-trivial, and in some cases is even infeasible [Ramsay and Silverman, 1997].

Nonethless, many interesting solutions can be found in the literature - such as the class of *thin-plate spline kernels* presented in [Duchon, 1977]. The thin-splate spline kernel defines a smooth hypersurface, with the norm of a function evaluating its curvature, e.g. for $m = 2$ and $\mathcal{X} = \mathbb{R}^2$:

$$\| f \|_{\mathcal{S}_2^2}^2 = \iint_{\mathcal{X}} \left\{ \left( \frac{\partial^2 f(\mathbf{x})}{\partial x_1^2} \right)^2 + \left( \frac{\partial^2 f(\mathbf{x})}{\partial x_1 \partial x_2} \right)^2 + \left( \frac{\partial^2 f(\mathbf{x})}{\partial x_2^2} \right)^2 \right\} \mathrm{d}x_1 \mathrm{d}x_2. \tag{2.19}$$

Note that the definition evaluates derivatives both along the axes *and* with respect to each variable. The optimal kernel function for this problem is given by the following expression, from [Poggio and Girosi, 1990]

$$\mathrm{K}(\mathbf{x}_i, \mathbf{x}_j) = \| \mathbf{x}_j - \mathbf{x}_i \|_2^2 \ln \left( \| \mathbf{x}_j - \mathbf{x}_i \|_2 \right). \tag{2.20}$$

## 2.3 Formulating Identification Problems with Kernels

So far, the discussion has focused on the construction of *functions* in an RKHS. But how does this relate to building *models*? This section will show how the results of the previous sections can be applied to nonlinear identification problems in practice.

### 2.3.1 Defining a Cost-Function

At this point, we know how to construct functions using kernels (2.4), and control the properties of these functions through the kernel (section 2.2). The task now is to estimate a function - which will be our model - from obtained measurements $\mathcal{Z}_n$.

Naturally, the first step in doing so is to define a cost-function (or optimisation criterion), such that estimated model will somehow resemble the data. Typically, this is done using a *loss-function*, i.e. a term penalising errors in the model, e.g.

$$\begin{aligned} \mathcal{J}_{\text{LS}}(f) &= \| y - f(x) \|_2^2 \\ &= \sum_{i=1}^{\text{N}} \Big( y_i - \langle f, k_{x_i} \rangle_{\mathcal{H}} \Big)^2 \\ &= \sum_{i=1}^{\text{N}} \Big( y_i - \sum_j \alpha_j k_{\mathbf{x}_j}(\mathbf{x}_i) \Big)^2. \end{aligned} \tag{2.21}$$

A model is obtained by minimising the optimisation criterion:

$$\hat{f} = \underset{f}{\operatorname{argmin}} \Big\{ \mathcal{J}_{ls}(f) \Big\}. \tag{2.22}$$

However, in a nonparametric setting, this problem is *ill-posed* [Hadamard, 1902] - i.e. there are an infinite number of possible solutions to the problem, all of which equally valid according to $\mathcal{J}_{ls}$. How to choose then ?

Typically, this is done by adding another constraint into the optimisation criterion, but this time solely constraining the properties of the model, e.g.

$$\begin{aligned} \mathcal{J}_{\text{RLS}}(f) &= \| y - f(x) \|_2^2 \; + \; \lambda \| f \|_{\mathcal{H}}^2 \\ &= \sum_{i=1}^{\text{N}} \Big( y_i - \langle f, k_{x_i} \rangle_{\mathcal{H}} \Big)^2 \qquad + \lambda \langle f, f \rangle_{\mathcal{H}} \\ &= \underbrace{\sum_{i=1}^{\text{N}} \Big( y_i - \sum_j \alpha_j k_{\mathbf{x}_j}(\mathbf{x}_i) \Big)^2}_{\text{error term}} \quad \underbrace{+ \lambda \sum_j \sum_k \alpha_j \alpha_k \text{K}(\mathbf{x}_j, \mathbf{x}_k)}_{\text{regularisation term}}. \end{aligned} \tag{2.23}$$

The term of the right-hand side is referred to as a *regularisation term*. Note that in the above expression, the points $\mathbf{x}_i$ refer to the N measured observations of the system, whereas the points $\mathbf{x}_j$ and $\mathbf{x}_k$ refer to the *a priori* infinite number of kernel slices required to define the function $f$ from the reproducing property (2.4).

Regularisation is an established statistical technique, both in the wider statistical community [Tikhonov and Arsenin, 1977], and also in system identification [Sjöberg et al., 1993, Chiuso, 2016]. In an RKHS sense, a regularisation term is a mathematical necessity to ensure the *well-posedness* of the solution, however its role can also be interpreted in terms of its effect on the model itself:

- Use of a regularisation term gives the user control over the *bias-variance* trade-off of the model. Increasing $\lambda$ places more weight on the model properties in the cost-function, reducing the sensitivity of the estimate to the data. However, controlling the variance in this way introduces a bias into the model, as now, with increasing $\lambda$, $\hat{f}$ moves further away from an interpolation of the data to a regression.

- However, in general this is desirable. For prediction, models that pure interpolations of the data are of little use, and acting on the variance in this way often improves the predictive capabilities of the model.
- The regularisation term ensures the well-posedness of the solution. In a direct sense, increasing $\lambda$ improves the numerical stability of the algorithm - which is closely linked with the idea of uniqueness.
- A regularisation term can also be thought of as a penalty on the model complexity. As, in a nonparametric setting, the complexity is not linked to the number of parameters, constraints are placed on the model as a whole. The cost-function now reads (from left to right) *'give me the most accurate model that is also simple, in the sense in which defined, to the degree specified by $\lambda$.'* This statement bears close resemblance to the aforementioned *parsimony principle* and *Occam's Razor*.
- Again, there is an equivalence between results formulated in the RKHS and other kernel-based paradigms. What is here referred to as a regularisation term becomes - in a Bayesian setting - a weight on the prior probability function, controlling the extent to which the distribution reflects the prior and the likelihood function.

All of these interpretations are linked, but however the role of the regularisation is interpreted one point is constant - the cost-function presented in (2.23) is defined *for a particular value of* $\lambda$, hence $\lambda$ must be chosen.

Methods for the selection of regularisation hyperparameters are widely discussed in the literature, with the most well-known being *cross-validation* [Wahba, 1990] and *marginal likelihood* [Rasmussen and Williams, 2006]. Other methods also exist, such as the L-Curve [Tenorio, 2001], but are less commonly used at present in system identification.

In this thesis, optimisation of hyperparameters will be performed exclusively using cross-validation (unless otherwise stated), as it is a reliable and well-understood method that provides a fair basis for comparison. That does not mean that investigation of other methods would not be interesting, just that it is not the principal focus of this work.

### 2.3.2 The Representer Theorem

Having discussed the role of the kernel function, the definition of a cost-function and the selection of a regularisation hyperparameter, the final step in identification process - prior to the estimation of the model parameters - is the definition of a model itself.

The expression of (2.4) is an infinite-dimensional expression for $f$ - which isn't much good to anyone. However, *the representer theorem* shows that the optimal model configuration for a cost-function such as $\mathcal{J}_{\text{RLS}}$ is a model constructed of a finite set of weighted kernels, centered at the observations:

$$\mathcal{F}_{\text{RLS}} : \quad f(\mathbf{x}) = \sum_{i=1}^{N} \alpha_i k_{\mathbf{x}_i}(\mathbf{x}), \quad \alpha \in \mathbb{R}^N. \tag{2.24}$$

Proof of this result can be found in different forms in various places in the literature, notably [Kimeldorf and Wahba, 1971] and [Schölkopf et al., 2001], however as it will aid the discussion of the following chapter, a summarised version of the proof presented in [Schölkopf et al., 2001] will be given here (though please refer to [Schölkopf et al., 2001] for a formal presentation).

***Proof of the Representer Theorem.*** For any $f \in \mathcal{H}$, it is possible to redefine $f$ as a function of two components :

$$f = f_{\parallel} + f_{\perp} \tag{2.25}$$

where $f_{\parallel}(z) = \langle f_{\parallel}, k_z \rangle_{\mathcal{H}_{\parallel}} = \sum_{i=1}^{N} \alpha_i k_{x_i}(z) \in \mathcal{H}_{\parallel}$ is a function spanning the finite-dimensional space of the observations and $f_{\perp}(z) = \langle f_{\perp}, k_z^* \rangle_{\mathcal{H}_{\perp}} = \sum_{j=1}^{\infty} \beta_j k_{x_i^*}^*(z) \in \mathcal{H}_{\perp}$ lies the part of $\mathcal{H}$ not spanned by the observations. These two components are orthogonal, such that :

$$\langle f_{\parallel}, f_{\perp} \rangle_{\mathcal{H}} = 0 \tag{2.26a}$$
$$\mathcal{H}_{\parallel} \cap \mathcal{H}_{\perp} = \varnothing \tag{2.26b}$$
$$\mathcal{H}_{\parallel} \cup \mathcal{H}_{\perp} = \mathcal{H} \tag{2.26c}$$

1. $f_{\perp}$ is irrelevant in the minimisation of the errors as :

$$\begin{aligned} f(\mathbf{x}) &= \langle f, k_{\mathbf{x}} \rangle_{\mathcal{H}} \\ &= \langle f_{\parallel} + f_{\perp}, k_{\mathbf{x}} \rangle_{\mathcal{H}} \\ &= \langle f_{\parallel}, k_{\mathbf{x}} \rangle + \langle f_{\perp}, k_{\mathbf{x}} \rangle_{\mathcal{H}} \\ &= \langle f_{\parallel}, k_{\mathbf{x}} \rangle_{\mathcal{H}}. \end{aligned} \tag{2.27}$$

2. The presence of $f_{\perp}$ ensures a non-minimal definition of $f$ as :

$$\begin{aligned} \| f \|_{\mathcal{H}}^2 &= \langle f, f \rangle_{\mathcal{H}} \\ &= \langle f_{\parallel} + f_{\perp}, f_{\parallel} + f_{\perp} \rangle_{\mathcal{H}} \\ &= \langle f_{\parallel}, f_{\parallel} \rangle_{\mathcal{H}} + \langle f_{\perp}, f_{\perp} \rangle_{\mathcal{H}} + \cancelto{0}{2\langle f_{\parallel}, f_{\perp} \rangle_{\mathcal{H}}} \\ &= \| f_{\parallel} \|_{\mathcal{H}}^2 + \| f_{\perp} \|_{\mathcal{H}}^2 \\ &\geq \| f_{\parallel} \|_{\mathcal{H}}^2, \end{aligned} \tag{2.28}$$

   with equality occurring if and only if $\| f_{\perp} \|_{\mathcal{H}}^2 = 0$.

□

The implication of this results is that $\mathcal{F}_{\text{RLS}}$ is the *optimal* model definition of $f$, for cost-functions such as $\mathcal{J}_{\text{RLS}}$. Therefore, neither adding terms to the representer nor changing the locations of kernels can improve the estimate - in fact it will adversely affect the results.

Whilst this results holds for both RKHS and Sobolev cases, in the Sobolev case, it is necessary to include a linear, parametric component in the model definition, corresponding to the null space of the derivative operator defined in the norm of the space. This term is often referred to in the literature as a *bias term*, and is a result of the boundary conditions of the Green's function. For example, in the case of the *spline-smoothing* problem defined in section 2.2.3, the optimisation function of (2.23) corresponds to

$$\begin{aligned} \mathcal{J}_{\mathcal{S}_m^2}(f) &= \| y - f(x) \|_2^2 + \lambda \| f \|_{\mathcal{S}_m^2}^2 \\ &= \sum_{i=1}^{N} \big( y_i - f(x_i) \big)^2 + \lambda \int_{\mathcal{X}} \left( \frac{\mathrm{d}^m f(x)}{\mathrm{d}x^m} \right)^2 \mathrm{d}x. \end{aligned} \tag{2.29}$$

for which a representer can be defined as

$$\mathcal{F}_{\text{SS}} : \quad f(\mathbf{x}) = \sum_{i=1}^{m} \theta_i x^{i-1} + \sum_{j=1}^{N} \alpha_j k_{x_j}(x), \quad \theta \in \mathbb{R}^m, \, \alpha \in \mathbb{R}^N. \tag{2.30}$$

### 2.3.3 Estimating the Model Parameters

With all the pieces of the puzzle now present, a solution for the optimal model parameters can be defined. It is trivial to show that, for the problem of $\mathcal{J}_{\text{RLS}}$ (2.23), with a representer defined according to $\mathcal{F}_{\text{RLS}}$ (2.24), the optimal model parameters can be estimated using the following expression, obtained by minimising $\mathcal{J}_{\text{RLS}}$ with respect to $\alpha$:

$$(\mathbf{K} + \lambda \mathbf{I})\, \alpha = \mathbf{y}. \tag{2.31}$$

for $\mathbf{y} = [y_1 \ \dots \ y_{\text{N}}]^\top \in \mathbb{R}^{\text{N}}$ and $\mathbf{K}, \mathbf{I} \in \mathbb{R}^{\text{N}\times\text{N}}$, subject to the choice of $\mathbf{K}$ and $\lambda$.

Similarly, the solution for the optimal parameters using $\mathcal{F}_{\text{SS}}$ (2.30) is given by the expression:

$$\begin{bmatrix} \mathbf{0} & \mathbf{X}^\top \\ \mathbf{X} & \mathbf{K} + \lambda \mathbf{I} \end{bmatrix} \begin{bmatrix} \theta \\ \alpha \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{y} \end{bmatrix}, \tag{2.32}$$

for $\mathbf{X} \in \mathbb{R}^{\text{N}\times m}$. The derivation here is less trivial, but is presented in [Wahba, 1990]. As mentioned, a linear term is included in the model definition to compensate for the null space of the derivative operator penalised in the optimisation criterion. However, as can be seen, this solution also corresponds with examples of *partially linear* and *semi-parametric* models given in the literature e.g. [Schölkopf et al., 2001, Espinoza et al., 2005].

## 2.4 The Smoothing Equivalence

In this section, a simulation example will be used to compare the two methods discussed up to this point (the *RKHS* and *Sobolev Space* methods). The aim of this example is to show that, despite the two methods acting on the model in different ways, the model properties can be effectively controlled either through the kernel hyperparameter or through a smoothing regulariser: with very similar results in each case.

### 2.4.1 The Data-Generating System

To illustrate this point, the example of a one-dimensional static nonlinear system is used. The system in question is nonsmooth, and therefore will be *outside* the model class of each method. It can be expected that this will naturally induce a bias into each of the models, and examination of how this bias manifests itself will allow us to compare the smoothing effect of each method:

$$\begin{aligned} \mathcal{S}_o : \ y(x_k) &= f_{\text{o}}(x_k) + e_{\text{o},k}, \quad x \in \mathbb{R} \\ f_{\text{o}}(x_k) &= \begin{cases} 1 - |x_k|, & |x_k| > 0.5 \\ x_k, & |x_k| \leq 0.5 \end{cases} \end{aligned} \tag{2.33}$$

It is considered that N = 200 observations, uniformly distributed across the input space ($x_i \sim \mathcal{U}(-1, 1)$), are available for estimation and validation of the models. And, these observations are highly noisy, with measurement corrupted at the output by white Gaussian noise of SNR = 5dB, where $\text{SNR} = 20 \log \left( \sigma_{f_{\text{o}}} / \sigma_e \right)$.

### 2.4.2 Identification Approaches

Two distinct approaches to the modelling problem were considered :

1. $\mathcal{M}_{\text{GRBF}}$: an 'RKHS' approach, using a Gaussian RBF kernel and functional norm regularisation. Here, the smoothness of the model is tuned through the kernel width $\sigma$, and the strength of the regularisation controlled through the regularisation hyperparameter $\lambda$. The model parameters were estimated using the expression of (2.31).
2. $\mathcal{M}_{\text{CSK}}$: a 'Sobolev Space' approach, using a cubic spline kernel. In this case there is no kernel hyperparameter, and the only hyperparameter to be tuned is the regularisation hyperparameter: which controls both the smoothness and the bias-variance trade-off of the model. As described in section 2.2.3, a bias function is also included in the model. In this case, the bias function is a linear term ($f_{\text{LIN}}(x) = \sum_{i=1}^{2} \theta_i x^{i-1}$), and thus, the corresponding model parameters were estimated using (2.32).

To estimate the models in each case, a two-step procedure was used:

1. The optimal model configuration was determined using a validation set (or cross-validation) approach, as described in [James et al., 2014], i.e. models were estimated using the training data and evaluated against a validation dataset generated under identical conditions. The optimal model configuration in each case is taken to be the one minimising the error (MSE) of the model with respect to the noisy validation data.
2. Following determination of the optimal model configuration, Monte-Carlo trials were run to test the performance of each algorithm. In this case $n_{mc} = 10^3$ trials were run, in which the noise-free training data was subject to different noise realisations.

In each case, the performance of the models was evaluated against a gridded dataset (with $n_{grid} = 10^3$), with the exception of the FIT value - which was evaluated against the noise-free validation data. The corresponding results are displayed in Table 2.1, calculated using the following expressions :

$$\begin{aligned}
\text{FIT} &= 100 \cdot \left(1 - \frac{\left\| y - \hat{f} \right\|_2^2}{\left\| y - \bar{y} \right\|_2^2}\right) \\
\text{MBIAS} &= \text{mean}_{i=1}^{n_{\text{GRID}}} \left\| y_{\text{o},i} - \text{mean}_{j=1}^{n_{\text{MC}}} \{\hat{f}_j(x_i)\} \right\|_1 \\
\text{MSDEV} &= \text{mean}_{i=1}^{n_{\text{GRID}}} \left\| \sigma_{\hat{f}}(x_i) \right\|_2 \\
\text{MRMSE} &= \text{mean}_{i=1}^{n_{\text{MC}}} \left\| y - \hat{f}_i \right\|_2 .
\end{aligned} \tag{2.34}$$

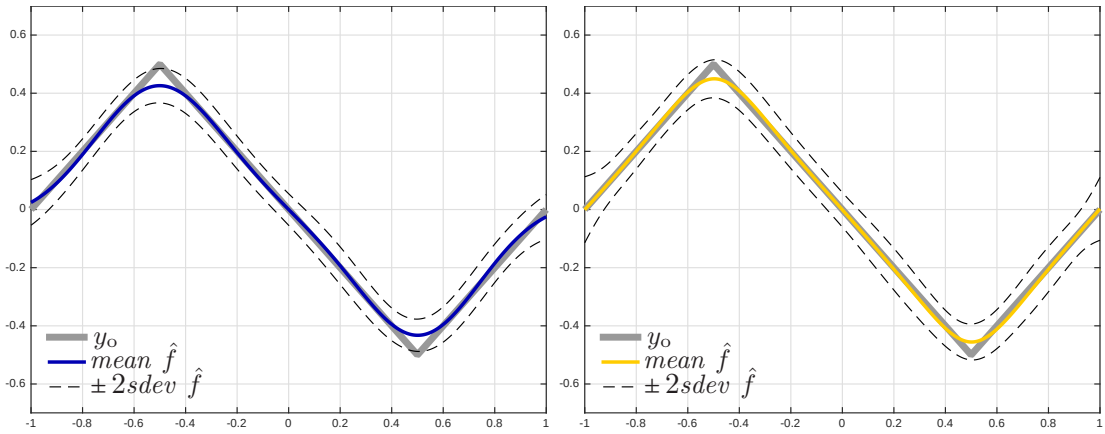
### 2.4.3 Optimal Results

| | FIT % | MBIAS | MSDEV | MRMSE | $\sigma$ | $\log_{10}\lambda$ |
|---|---|---|---|---|---|---|
| $\mathcal{M}_{\text{GRBF}}$ | 91.42 | .0345 | $3.38 \times 10^{-4}$ | .0300 | 0.31 | 0.27 |
| $\mathcal{M}_{\text{CSK}}$ | 88.92 | .0348 | $1.27 \times 10^{-4}$ | .0336 | N/A | −2.57 |

TABLE 2.1 – Summarised results of the optimised models in Figure 2.8.

Whilst this is a very simple example, it allows us to understand how the two approaches, $\mathcal{M}_{\text{GRBF}}$ and $\mathcal{M}_{\text{CSK}}$, control the model properties.

As can be seen from Table 2.1, both approaches perform similarly. $\mathcal{M}_{\text{GRBF}}$ is overall slightly better (from observation of the fitting scores and MRMSE), which can be partly attributed to the extra degree of freedom in the optimisation problem (two hyperparameters instead of one).




(a) $\mathcal{M}_{\text{GRBF}}$ (b) $\mathcal{M}_{\text{CSK}}$

FIGURE 2.8 – Optimal results.

Observations of the plots in Figure 2.8 support thesse statements: both models estimate well the true function, with comparable bias and variance in each case.

The differences in the two estimated models are less a feature of the algorithms, but rather the smoothness assumptions placed in each case. $\mathcal{M}_{\text{GRBF}}$ used a Gaussian RBF kernel, which acts as a smooth Gaussian filter, whereas $\mathcal{M}_{\text{CSK}}$ penalises the second derivative of the model: effectively acting as a *Butterworth* filter [Wahba, 1990].

In fact, visually the results of Figure 2.8b are arguably preferable to those of Figure 2.8a, as the smoothness assumption of $\mathcal{M}_{\text{CSK}}$ is only violated at two locations ($x = \{-0.5, 0.5\}$), and otherwise respected. However, the differences between the two approaches are minimal.

### 2.4.4 Tuning the Model Smoothness

We can also compare the smoothing effect of each approach - to better understand how the model properties are controlled in each case.

Here, $\mathcal{M}_{\text{GRBF}}$ was re-estimated over a range of $\sigma$ values, with the 'optimal' value of $\lambda$ estimated in the previous section ($\log_{10}\lambda = 0.27$). Similarly, $\mathcal{M}_{\text{CSK}}$ was re-estimated over a range of $\lambda$ values.

Figure 2.9 displays a selection of the results, chosen such that comparable levels of bias and variance are visible on the left and right-hand plots in each case. For reference, the chosen hyperparameter values are given in Table 2.2.

As can be seen in Figure 2.9, the effect of varying the hyperparameters in each case is extremely similar - both in the way smoothing the model acts on the variance and in the way smoothing the model acts on the mean. Again, the differences in the models relate more to the types of smoothness assumptions placed by the algorithms, rather than the way in which the properties are controlled.

| | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| $\sigma_{\text{GRBF}}$ | 0.044 | 0.20 | 1.11 | 2.21 | 6.16 |
| $\log_{10} \lambda_{\text{CSK}}$ | −6.01 | −4.00 | −1.31 | −0.28 | 1.69 |

TABLE 2.2 – Hyperparameter values for Figure 3.12.

This may appear somewhat strange - what is so interesting about two sets of results that are more or less the same ?

The answer is simple - what is interesting is the way in which these results have been obtained in each case. $\mathcal{M}_{\text{GRBF}}$ places a smoothness constraint on the model through the kernel function (i.e. through the hyperparameter $\sigma$) - which directly constrains the estimation model class. This is what we have referred to as a *hard bound*, as no matter what data is provided, or what level of regularisation is used, there is a strict constraint on the minimum smoothness (or maximum flexbility) of any model estimated using $\mathcal{M}_{\text{GRBF}}$ for a given value of $\sigma$.

By constrast, $\mathcal{M}_{\text{CSK}}$ defines an *a priori* flexible kernel function (in this case the cubic spline kernel). Here, the flexibility of the model class is related to the data (both the number of the observations available, and their distribution), however flexibility can be augmented by adding 'knots' to the model definition, for example as discussed in [Wahba, 1990]. A similar method, using added grid points, will be discussed in the following chapter.

Now, smoothness is controlled by varying $\lambda$ in the cost-function - which controls the extent to which the Sobolev norm of the estimated model is penalised. In the Sobolev case, a 'smaller' function is also a smoother function - hence, whilst the model class definition doesn't change (the kernel stays the same), the algorithm is able to act on the model properties through the parameters. This can be viewed as a constraint on the complexity of the model, whereby 'simplicity' is defined as 'smoothness'.

Whilst the value of $\lambda$ does tell us something about the smoothness of the model ($\lambda = 0$ gives an interpolation, and as $\lambda \to \infty$, the model tends to the bias functions), however the smoothness of the model is also dependant on the data. Hence, this is referred to here as a *soft bound* - as $\lambda$ does not place hard bounds on the model class in the same way as $\mathbf{K}$.

## 2.5 Summary

In this chapter, a review of RKHS and Sobolev space methods for nonlinear identification was presented, with emphasis on the impact of the kernel selection and the regularisation on the model class definition.

The notions of hard and soft model constraints were introduced. Hard constraints, placed through the choice of kernel function, allow strict bounds on the model class to be placed : enabling the application of kernel methods to problems of interest in both the control and identification communities. Soft model constraints, placed through a regularisation term, allow continuous complexity tuning as required and open up new possibilities in how the model class can be tuned.

Using a simple example, it was shown how a soft optimisation over the model properties can achieve equivalent control with respect to kernel selection over features such as smoothness, by

using a flexible kernel definition and a feature-oriented regularisation term.

However, the Sobolev space falls down in two particular areas. Whilst the ability to control features in the model through a regularisation hyperparameter is very attractive, the extent to which this can be used is limited by the need to formulate an solve the Green's function for the corresponding problem. As such, tuning arbitrary features in the model may be challenging in specific problems, for which the correct solutions are unknown. Similarly, integrating such constraints into different nonlinear problems is also potentially tricky: as the approach relies on the definition of a general, flexible kernel function from which the properties can be controlled.

The following chapter will aim to rectify this by formulating an approach that allows for the integration of both hard and soft constraints into the optimisation scheme as required, in a formulation consistent with the RKHS approach presented in section 2.3. By placing constraints directly in a specified RKHS, rather than specifying an RKHS in terms of the properties of interest, arbitrary constraints can be defined as desired using differential operators and incorporated into different types of kernel functions and structures as required.

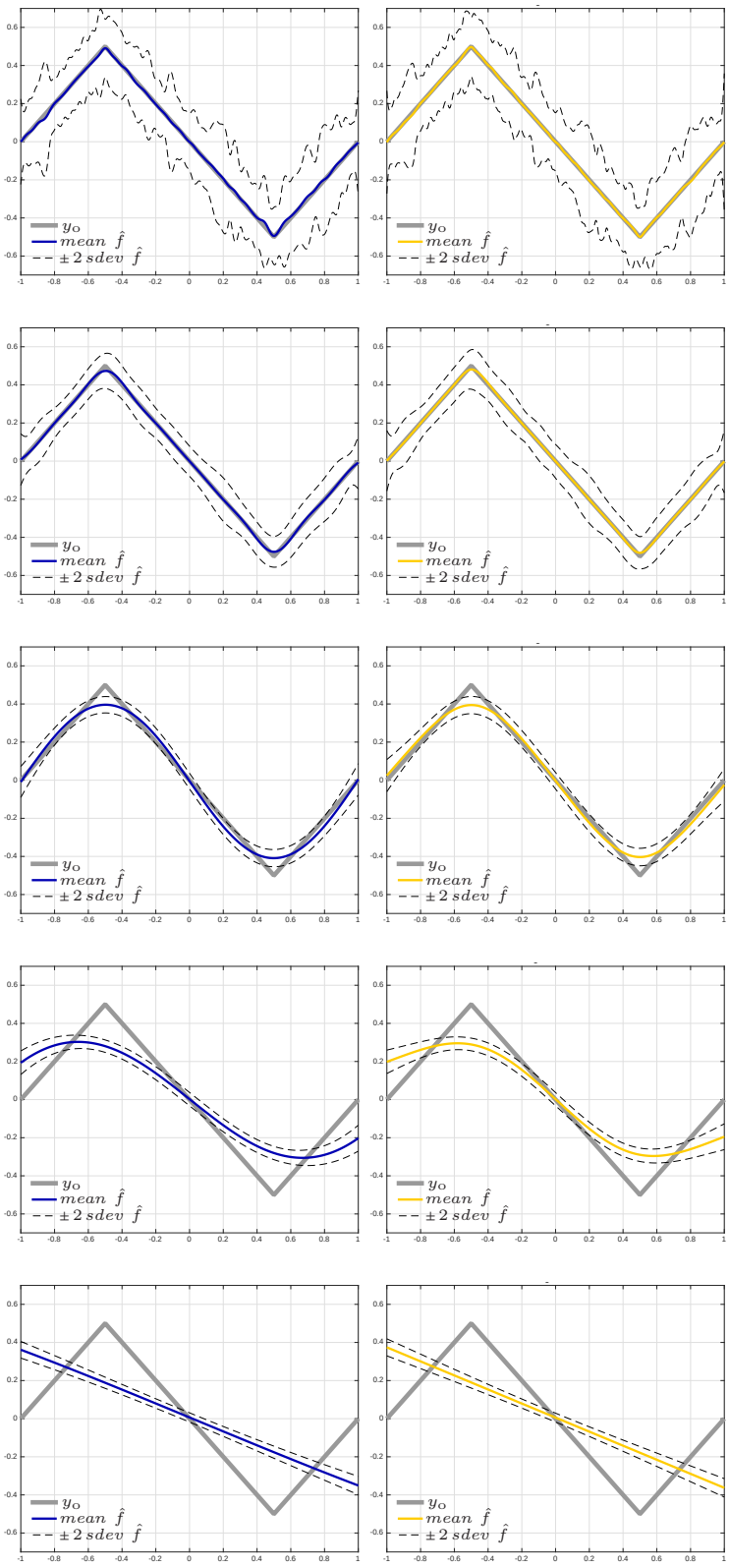



FIGURE 2.9 – Comparing the smoothing effect of tuning the kernel hyperparameter in (2.23) (left) with the regularisation hyperparameter in (2.29) (right).

# 3

# Penalising Derivatives in the RKHS

*This chapter will present a detailed analysis of two methods for constraining functional derivatives in RKHS system identification. The first is a method taken from the literature which will be referred to here as the 'indirect method', as it constrains evaluations of functional derivatives. The second method is a novel contribution, and is nominally termed the 'direct method' as it allows derivatives of functions to be penalised directly in an RKHS - in a similar manner to the methods presented in the previous chapter.*

## 3.1 Introduction

In the previous chapter, two methods for kernel-based identification were presented: RKHS methods and Sobolev space methods. As discussed, there is a strong link betweeen the two approaches - in the formulation considered the Sobolev space is also an RKHS with an associated reproducing kernel, but with certain additional properties.

These differences in fact represent an important paradigm shift. In conventional RKHS methods, the properties of the model are tuned primarily through the kernel, effectively placing hard bounds on the model class, with a regularisation term used to ensure the well-posedness of the solution. By contrast, in a Sobolev space formulation an *a priori* kernel function is specified, and soft bounds are placed on the model properties through a regularisation term. This arises because the Sobolev norm is defined with respect to derivatives of the function, hence constraining the norm of $f$ also acts on these properties.

This formulation has been mostly explored in the context of smoothing regularisers. Interestingly, as shown in the previous chapter, such an approach can achieve very similar levels of control over the model properties as a kernel-based, despite not acting on the model class. This is referred to in the previous chapter as 'the smoothing equivalence'. This opens up many possibilities with respect to how the properties of a model could be controlled in nonlinear identification.

However, as will be discussed in section 3.2, derivatives of functions are already well-defined in an RKHS. Which means that properties encoded into the specification of a Sobolev space can be expressed directly in an RKHS.

This has already been exploited to a certain degree in the literature, with examples existing of derivatives being used in a diverse range of problems. For example, scope exists in this scenario for

nonparametric methods to be used a method for solving linear and nonlinear ordinary differential equations (see e.g. [Mehrkanoon et al., 2012]). Also, this permits their application to the problem of 'learning from derivative observations' [Solak et al., 2003, Zhou, 2008, Kondor et al., 2005], for example the problem of accurately determining position and velocity for a device based on acceleration measurements. This problem has itself been adapted to allow derivative properties to be constrained, notably with application to variable selection (e.g. [Rosasco et al., 2010, Duijkers et al., 2014]). This will be discussed in section 3.3.

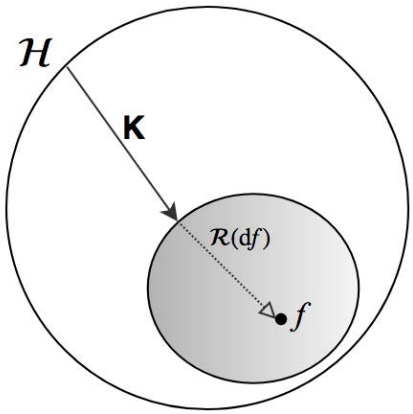



FIGURE 3.1 – Formulating derivative penalities in an *RKHS* directly allows **hard** and **soft** optimisation of the model class.

The major contribution of this thesis is to add to the current literature by formulating and analysing a method for direct penalisation of derivative properties in an RKHS - allowing soft control over the model properties through a regularisation term, and hard control through the choice of kernel function (as depicted in Figure 3.1). Examples of such an approach can be found in the literature (e.g. [Lauer et al., 2012]), however to the best of the author's knowledge a full treatment of the method from a theoretical perspective and a practical perspective has not been presented. Hence, section 3.4 will formulate an optimisation scheme and solution, in addition to discussing the choice of representer, the role of the bias function and the kernel selection.

Importantly, this approach is based on a theoretical approximation in the definition of the representer. By contrast, the Sobolev space is a theoretically exact approach to same problem. However, it will be shown in section 3.5 that - subject to proper selection of the kernel function - equivalent results can be obtained using either method.

## 3.2 Derivatives in the RKHS

In this section, we will introduce several results and some notation to facilitate the discussion of the following sections. The results follow from the work of [Zhou, 2008], examining the use of derivative operators in the RKHS.

### 3.2.1 Derivatives in One Dimension

Differentiation can be thought of as a linear operator since (by analogy with (1.2)):

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}x}\Big\{ f(x) + g(x) \Big\} &= \frac{\mathrm{d}}{\mathrm{d}x}\Big\{ f(x) \Big\} + \frac{\mathrm{d}}{\mathrm{d}x}\Big\{ g(x) \Big\}, \\ \text{and} \qquad \frac{\mathrm{d}}{\mathrm{d}x}\Big\{ c\, f(x) \Big\} &= c\, \frac{\mathrm{d}}{\mathrm{d}x}\Big\{ f(x) \Big\}. \end{aligned} \tag{3.1}$$

In the above expressions, $f : \mathbb{R} \to \mathbb{R}$ and $g : \mathbb{R} \to \mathbb{R}$ are arbitrary, differentiable functions, and $c \in \mathbb{R}$ is a constant scalar. As such, derivatives can be explicitly formulated in an RKHS, and the expressions of section 2.2.1 relating functions to kernels can be extended to relate derivatives of functions to derivatives of kernels.

For example, consider the simple case $f \in \mathcal{H}$ for $\mathcal{X} \in \mathbb{R}$. Let $\mathcal{H}$ be a $C^k$ continuous functional space, from [Zhou, 2008],

$$\frac{\mathrm{d}^m f}{\mathrm{d}x^m} \in \mathcal{H} \quad \text{if } 2m \leq k. \tag{3.2}$$

This permits the definition of a *derivative reproducing property*, equivalent to (2.4),

$$\begin{aligned} \frac{\mathrm{d}^m}{\mathrm{d}x^m}\Big\{ f(x) \Big\} &= \Big\langle f, \frac{\mathrm{d}^m}{\mathrm{d}x^m}\Big\{ k_x \Big\} \Big\rangle_{\mathcal{H}} \\ &= \sum_i \alpha_i \frac{\mathrm{d}^m}{\mathrm{d}x^m}\Big\{ k_{x_i}(x) \Big\}. \end{aligned} \tag{3.3}$$

Using (3.3), the derivatives of a function can be computed exactly, subject to the definition of the corresponding derivative kernel. To illustrate this result, we will compute the derivatives of a Gaussian RBF kernel.

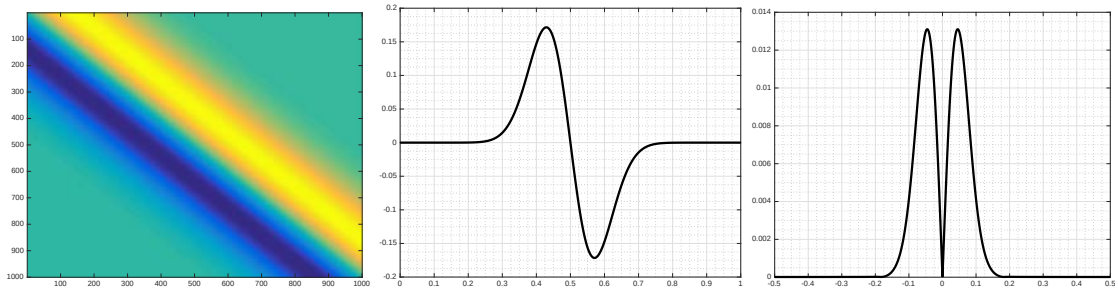
$$\frac{\mathrm{d}}{\mathrm{d}x}\Big\{ k_{x_i}(x) \Big\} = \frac{2}{\sigma^2}(x - x_i) \exp\left\{ -\frac{\|x - x_i\|^2}{\sigma^2} \right\}. \tag{3.4}$$

$$\frac{\mathrm{d}^2}{\mathrm{d}x^2}\Big\{ k_{x_i}(x) \Big\} = -\frac{2}{\sigma^2}\Big[ 1 - \frac{2}{\sigma^2}(x - x_i)^2 \Big] \exp\left\{ -\frac{\|x - x_i\|^2}{\sigma^2} \right\}. \tag{3.5}$$

The Gaussian RBF kernel is a very nice kernel to work with in this case: it is infinitely differentiable, it's derivatives are known (the Hermite polynomials), and they are also easy to compute for multidimensional kernels (derivatives are sums and products of partial derivatives). Figures 3.2 and 3.3 illustrate the first and second derivatives of the Gaussian kernel. As can be seen in Figures 3.2c and 3.3c, the derivatives of the kernel are band-pass filters - with the band controlled by width of the kernel and the order of the derivative [Romeny, 2008]. Increasing the order of the derivative pushes the pass band 'outwards', towards higher-frequency components.

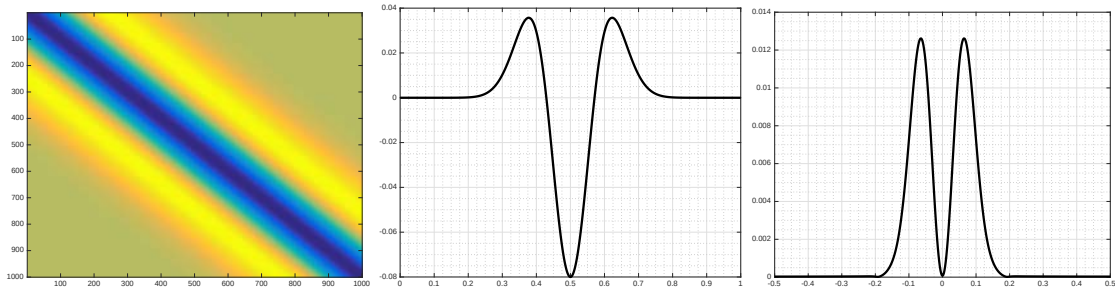
The norm of a functional derivative can also be computed, in a similar fashion to (2.6).

$$\begin{aligned} \left\| \frac{\mathrm{d}^m f}{\mathrm{d}x^m} \right\|_{\mathcal{H}}^2 &= \Big\langle \frac{\mathrm{d}^m f}{\mathrm{d}x^m}, \frac{\mathrm{d}^m f}{\mathrm{d}x^m} \Big\rangle_{\mathcal{H}} \\ &= \Big\langle \sum_i \alpha_i \frac{\mathrm{d}^m}{\mathrm{d}x^m}\Big\{ k_{x_i} \Big\}, \sum_j \alpha_j \frac{\mathrm{d}^m}{\mathrm{d}x^m}\Big\{ k_{x_j} \Big\} \Big\rangle_{\mathcal{H}} \\ &= \sum_i \sum_j \alpha_i \alpha_j \frac{\partial^{2m} \mathrm{K}(x_i, x_j)}{\partial x_i^m \, \partial x_j^m}. \end{aligned} \tag{3.6}$$



(a) $\mathcal{D}^{(0,1)}\mathbf{K}$ (b) $\mathcal{D}^{(0,1)}\mathrm{K}(\cdot,\mathbf{x})$ (c) $\mathcal{F}\{\mathcal{D}^{(0,1)}\mathrm{K}(\cdot,\mathbf{x})\}$

FIGURE 3.2 – The first derivative of the Gaussian radial basis function kernel in one dimension.



(a) $\mathcal{D}^{(0,2)}\mathbf{K}$ (b) $\mathcal{D}^{(0,2)}\mathrm{K}(\cdot,\mathbf{x})$ (c) $\mathcal{F}\{\mathcal{D}^{(0,2)}\mathrm{K}(\cdot,\mathbf{x})\}$

FIGURE 3.3 – The second derivative of the Gaussian radial basis function kernel in one dimension.

Here, derivatives are evaluated with respect to the left-hand and right-hand sides of the kernel, e.g.

$$\frac{\partial^2 \mathrm{K}(x_i,x_j)}{\partial x_i\,\partial x_j} = \frac{2}{\sigma^2}\Big[\,1 - \frac{2}{\sigma^2}(x_j - x_i)^2\,\Big]\exp\left\{-\frac{\|x_j - x_i\|^2}{\sigma^2}\right\}. \tag{3.7}$$

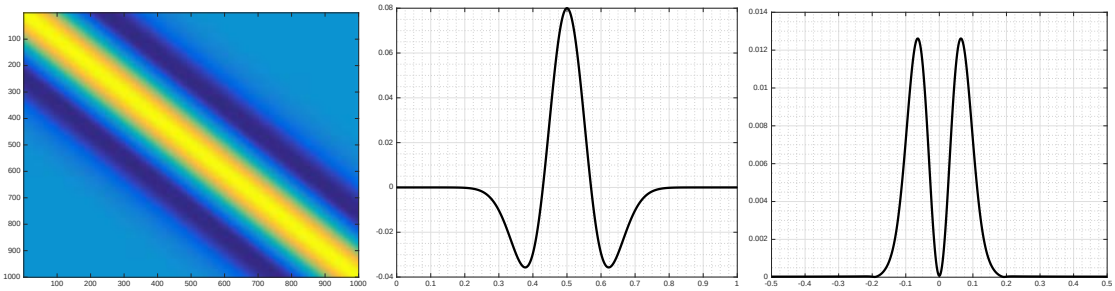
$$\frac{\partial^4 \mathrm{K}(x_i,x_j)}{\partial x_i^2\,\partial x_j^2} = \frac{4}{\sigma^4}\Big[\,3 - \frac{12}{\sigma^2}(x_j - x_i)^2 + \frac{4}{\sigma^4}(x_j - x_i)^4\,\Big]\exp\left\{-\frac{\|x_j - x_i\|^2}{\sigma^2}\right\}. \tag{3.8}$$

Figures 3.4 and 3.5 illustrate the kernels of (3.7). As can be seen, in the case of the Gaussian kernel, Figure 3.4b is negation of Figure 3.3b. This is not a general result, but does apply to certain classes of kernel (see for example [Kondor et al., 2005]).

### 3.2.2 Derivatives in Multiple Dimensions

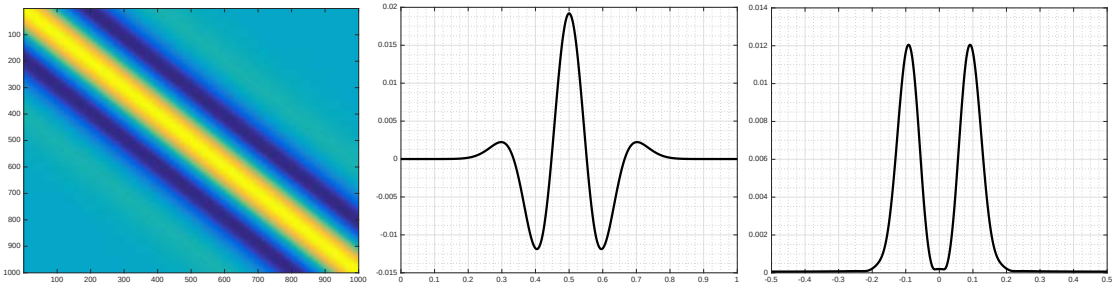
The results of the previous section can be extended to any $n_x$-dimensional space ($\mathcal{X} = \mathbb{R}^{n_x}$), subject to the same conditions as (3.2),

$$\partial_{x_i}^{m_i} f \in \mathcal{H} \text{ if } 2m_i < k. \tag{3.9}$$



(a) $\mathcal{D}^{(1,1)}\mathbf{K}$ (b) $\mathcal{D}^{(1,1)}\mathrm{K}(\cdot,\mathbf{x})$ (c) $\mathcal{F}\{\mathcal{D}^{(1,1)}\mathrm{K}(\cdot,\mathbf{x})\}$

FIGURE 3.4 – The norm of the first derivative of the Gaussian radial basis function kernel in one dimension.



(a) $\mathcal{D}^{(2,2)}\mathbf{K}$ (b) $\mathcal{D}^{(2,2)}\mathrm{K}(\cdot,\mathbf{x})$ (c) $\mathcal{F}\{\mathcal{D}^{(2,2)}\mathrm{K}(\cdot,\mathbf{x})\}$

FIGURE 3.5 – The norm of the second derivative of the Gaussian radial basis function kernel in one dimension.

To simplify the exposition of the coming section, the following shorthand notation is now introduced.

$$\begin{aligned}
\mathcal{D}^{\mathbf{m}}(\,\cdot\,) &= \partial_{x_1}^{m_1}\partial_{x_2}^{m_2}\,\ldots\,\partial_{x_{n_x-1}}^{m_{n_x-1}}\partial_{x_{n_x}}^{m_{n_x}}(\,\cdot\,)\\
\mathcal{D}^{(\mathbf{m},\mathbf{m})}(\,\cdot\,) &= \partial_{x_1}^{(m_1,m_1)}\,\ldots\,\partial_{x_{n_x}}^{(m_{n_x},m_{n_x})}(\,\cdot\,)\\
\partial_{x_i}^{m_i}(\,\cdot\,) &= \frac{\partial^{m_i}(\,\cdot\,)}{\partial {x_i}^{m_i}}\\
\partial_{x_i}^{(m_i,m_i)}(\,\cdot\,) &= \partial_{x_i}^{m_i}\partial_{x_i^\star}^{m_i}(\,\cdot\,) = \frac{\partial^{2m_i}}{\partial x_i^{m_i}\partial {x_i^\star}^{m_i}}(\,\cdot\,), \quad \mathbf{m}\in\mathbb{N}_0^{n_x}.
\end{aligned} \tag{3.10}$$

Using these operators and the results presented in [Zhou, 2008], a general *derivative reproducing property* can be written as

$$\begin{aligned}
\mathcal{D}^{\mathbf{m}} f(\mathbf{x}) &= \langle f, \mathcal{D}^{\mathbf{m}} k_{\mathbf{x}}\rangle_{\mathcal{H}}\\
&= \langle f, \partial_{x_1}^{m_1}\partial_{x_2}^{m_2}\,\ldots\,\partial_{x_{n_x-1}}^{m_{n_x-1}}\partial_{x_{n_x}}^{m_{n_x}}\,k_{\mathbf{x}}\rangle_{\mathcal{H}}.
\end{aligned} \tag{3.11}$$

Similarly, an inner-product can be defined for derivatives of kernels :

$$\begin{aligned}
\mathcal{D}^{(\mathbf{m},\mathbf{m})}\mathbf{K} &= \langle \mathcal{D}^{\mathbf{m}} k_{\mathbf{x}}, \mathcal{D}^{\mathbf{m}} k_{\mathbf{x}'}\rangle_{\mathcal{H}}\\
&= \langle \partial_{x_1}^{m_1}\,\ldots\,\partial_{x_{n_x}}^{m_{n_x}}\,k_{\mathbf{x}}, \partial_{x'_1}^{m_1}\,\ldots\,\partial_{x'_{n_x}}^{m_{n_x}}\,k_{\mathbf{x}'}\rangle_{\mathcal{H}}\\
&= \partial_{x_1}^{(m_1,m_1)}\,\ldots\,\partial_{x_{n_x}}^{(m_{n_x},m_{n_x})}\,\mathbf{K}.
\end{aligned} \tag{3.12}$$

### 3.2.3 Examples of Derivative Operators

Using the results of the previous section, many different types of derivative operators can be defined and evaluated using kernels. To illustrate this, two examples are given here - a smoothness operator and an interation operator.

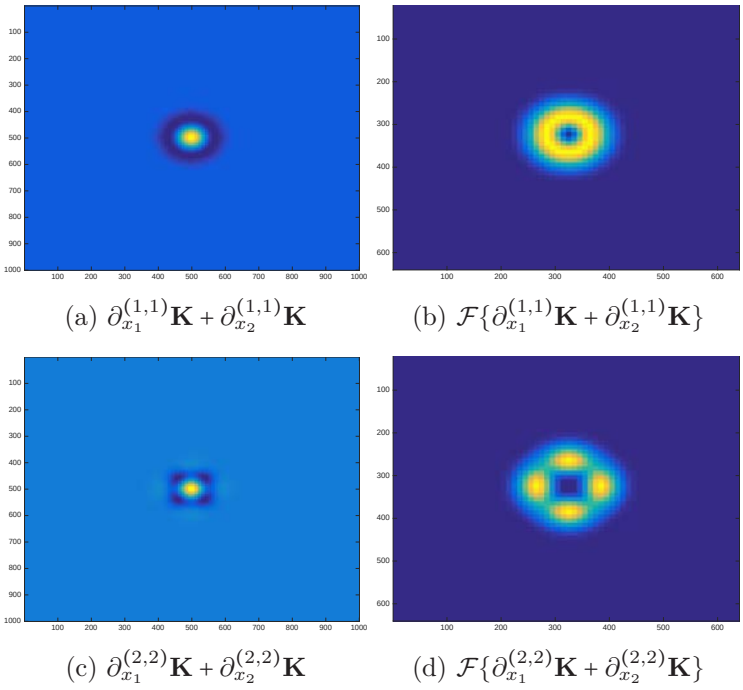
We will consider the operators on a space defined in $\mathcal{X} = \mathbb{R}^2$, to show in particular, how one of the most interesting aspects of using differential operators is that, by combining derivatives in different ways, very different properties can be evaluated.

**Case 1: The Smoothness Operator**

A smoothness operator can be defined by forming a vector of partial derivatives in each direction. The norm in this case can be evaluated as a *sum of the norms* of partial derivatives of $f$,

$$\sum_{i=1}^{n_x} \left\| \partial_{x_i}^m f \right\|_{\mathcal{H}}^2 = \sum_i \sum_j \alpha_i \alpha_j \underbrace{\left( (\partial_{x_1}^{(m,m)} + \partial_{x_2}^{(m,m)}) \mathrm{K}(\mathbf{x}_i, \mathbf{x}_j) \right)}_{\text{Figure 3.6}}. \tag{3.13}$$

In Figure 3.6, (3.13) is plotted over values evaluated at the origin and a two dimensional grid, for $m = 1$ (i.e. the gradient operator) and $m = 2$.



(a) $\partial_{x_1}^{(1,1)}\mathbf{K} + \partial_{x_2}^{(1,1)}\mathbf{K}$ (b) $\mathcal{F}\{\partial_{x_1}^{(1,1)}\mathbf{K} + \partial_{x_2}^{(1,1)}\mathbf{K}\}$

(c) $\partial_{x_1}^{(2,2)}\mathbf{K} + \partial_{x_2}^{(2,2)}\mathbf{K}$ (d) $\mathcal{F}\{\partial_{x_1}^{(2,2)}\mathbf{K} + \partial_{x_2}^{(2,2)}\mathbf{K}\}$

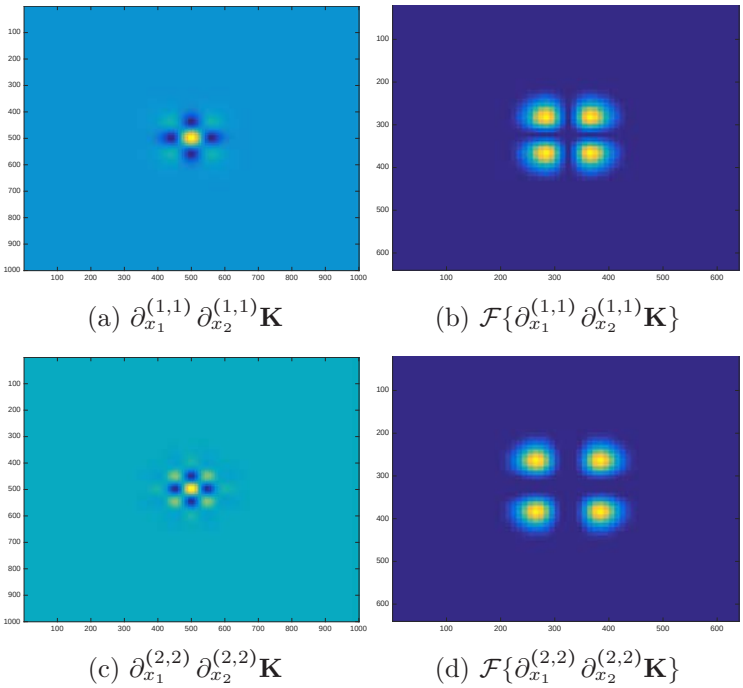
FIGURE 3.6 – *Smoothness* kernels can be formed by taking sums of partial derivatives.

**Case 2: The Interaction Operator**

Interestingly, combining partial derivatives allows interactions between variables to be evaluated. Again, a norm for such an operator can be evaluated using the expressions of the previous

section,

$$\left\| \partial_{x_1}^m \partial_{x_2}^m f \right\|_{\mathcal{H}}^2 = \sum_i \sum_j \alpha_i \alpha_j \underbrace{\left( \partial_{x_1}^{(m,m)} \partial_{x_2}^{(m,m)} \mathrm{K}(\mathbf{x}_i, \mathbf{x}_j) \right)}_{\text{Figure 3.7}}. \tag{3.14}$$



(a) $\partial_{x_1}^{(1,1)} \partial_{x_2}^{(1,1)} \mathbf{K}$ (b) $\mathcal{F}\{\partial_{x_1}^{(1,1)} \partial_{x_2}^{(1,1)} \mathbf{K}\}$

(c) $\partial_{x_1}^{(2,2)} \partial_{x_2}^{(2,2)} \mathbf{K}$ (d) $\mathcal{F}\{\partial_{x_1}^{(2,2)} \partial_{x_2}^{(2,2)} \mathbf{K}\}$

FIGURE 3.7 – *Interaction* kernels can be formed by taking products of partial derivatives.

Figure 3.7 plots the kernels in the same manner as Figure 3.6. Comparison of the two sets of plots clearly how they differ. In particular, Figure 3.6d shows that the smoothness operator primarily focuses on components along the axis, whereas Figure 3.7d exclusively focuses on components along diagonals - i.e. between variables.

And, these two constraints can be combined to form an isotropic penalty - as in the *thin-plate splines* case.

## 3.3 The Indirect Approach

Hence, if derivatives are well-formulated in an RKHS, they can be incorporated into the optimisation criterion. The first of the two methods presented in this chapter considers a penalty on evaluations of functional derivatives, i.e. derivatives are formulated in $\mathcal{H}$, but penalised in $\ell_2$.

This approach is essentially equivalent to the problem of *'learning with derivatives'*, which has been discussed in the literature from both an RKHS perspective [Kondor et al., 2005, Zhou, 2008] and a Gaussian Process perspective [Solak et al., 2003, Rasmussen and Williams, 2006].

Adapting this problem to allow derivative properties to be constrained can be done simply by neglecting consideration of observations but still penalising derivatives. Such an approach

has been explored in the literature, notably in [Rosasco et al., 2010] for variable selection and [Duijkers et al., 2014] for order selection in LPV models.

This approach is referred to here as the *indirect approach* as, unlike in the case of (2.29), the constraint is not placed over the global behaviour of the model - but instead only locally.

This has several major implications - both positive and negative:

- Firstly, it allows a theoretically exact formulation in an RKHS setting, therefore the user is still afforded flexibility with respect to how the kernel function can be defined.
- Secondly, in this formulation there is no need for a bias function.
- And, penalising evaluations of $f$ in $\mathbb{R}$ gives flexibility in how such constraints are penalised. Here, the smooth constraints (in $\ell_2$) are considered, but the theory extends to other types of penalties (e.g. $\ell_1$ or $\ell_\infty$).
- However, penalising derivatives in this way by necessity requires additional terms in the model definition. This results in a more convoluted formulation than the two approaches presented so far, with it becoming increasingly complicated for every additional term added. And even in the case of a single constraint: the model definition changes for every problem.

### 3.3.1 Penalising Evaluations of Derivatives in the RKHS

An indirect approach for derivative regularisation can be formulated as follows:

$$\begin{aligned}\mathcal{J}_{\text{DMIN}}(f) &= \|\mathbf{y} - \mathbf{f}(\mathbf{x})\|_2^2 + \gamma \,\|\, \mathcal{D}^{\mathbf{m}}\mathbf{f}(\mathbf{x}) \,\|_2^2 + \lambda \| f \|_{\mathcal{H}}^2 \\ &= \sum_{i=1}^{N} \left\{ (y_i - \langle f, k_{x_i} \rangle_{\mathcal{H}})^2 + \gamma \, (0 - \langle f, \mathcal{D}^{\mathbf{m}} k_{x_i} \rangle_{\mathcal{H}})^2 \right\} + \lambda \langle f, f \rangle_{\mathcal{H}}.\end{aligned} \tag{3.15}$$

Note that here it is considered that constraints are placed at the observations, but in fact the locations are unconstrained [Zhou, 2008].

Although this formulation represents a slight deviation from what we would like to achieve, it permits the definition of an optimal model for problems of this form, according to :

$$\mathcal{F}_{\text{DMIN}} \quad f(\mathbf{x}) = \sum_{i=1}^{N} \alpha_i k_{\mathbf{x}_i}(\mathbf{x}) + \sum_{i=1}^{N} \beta_i \, \mathcal{D}^{\mathbf{m}} k_{\mathbf{x}_i}(\mathbf{x}). \tag{3.16}$$

The proof of this bears analogy to the proof of the representer for $\mathcal{F}_{\text{RLS}}$, and can be found in [Zhou, 2008]. To minimise observations of derivatives it is necessary to include any and all additionally constrained terms in the representer of $f$.

### 3.3.2 Determining the Optimal Model Configuration

A solution for the model parameters $\alpha$, $\beta$ can be obtained by minimising with respect to the parameters, which yields the following solution:

$$(\mathbf{A} + \gamma\mathbf{B} + \lambda\mathbf{C}) \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \mathbf{K}^\top \\ \mathcal{D}^{(0,\mathbf{m})}\mathbf{K}^\top \end{bmatrix} \mathbf{y}, \tag{3.17}$$

subject to the definition of matrices $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ :

$$\mathbf{A} = \begin{bmatrix} \mathbf{K}^\top\mathbf{K} & \mathbf{K}^\top\mathcal{D}^{(0,\mathbf{m})}\mathbf{K} \\ \mathcal{D}^{(0,\mathbf{m})}\mathbf{K}^\top\mathbf{K} & \mathcal{D}^{(0,\mathbf{m})}\mathbf{K}^\top\mathcal{D}^{(0,\mathbf{m})}\mathbf{K} \end{bmatrix} \tag{3.18a}$$

$$\mathbf{B} = \begin{bmatrix} \mathcal{D}^{(0,\mathbf{m})}\mathbf{K}^\top\mathcal{D}^{(0,\mathbf{m})}\mathbf{K} & \mathcal{D}^{(0,\mathbf{m})}\mathbf{K}^\top\mathcal{D}^{(0,2\mathbf{m})}\mathbf{K} \\ \mathcal{D}^{(0,2\mathbf{m})}\mathbf{K}^\top\mathcal{D}^{(0,\mathbf{m})}\mathbf{K} & \mathcal{D}^{(0,2\mathbf{m})}\mathbf{K}^\top\mathcal{D}^{(0,2\mathbf{m})}\mathbf{K} \end{bmatrix} \tag{3.18b}$$

$$\mathbf{C} = \begin{bmatrix} \mathbf{K} & \mathcal{D}^{(0,\mathbf{m})}\mathbf{K} \\ \mathcal{D}^{(\mathbf{m},0)}\mathbf{K} & \mathcal{D}^{(\mathbf{m},\mathbf{m})}\mathbf{K} \end{bmatrix}, \tag{3.18c}$$

where $\{\, \mathcal{D}^{(\mathbf{a},\mathbf{b})}\mathbf{K} \,\}_{i,j} = \mathcal{D}^{(\mathbf{a},\mathbf{b})}\mathrm{K}(\mathbf{x}_i, \mathbf{x}_j)$.

## 3.4 The Direct Approach

In the previous section, derivative constraints were incorporated into the cost-function as evaluations of derivatives - which permits a theoretically exact solution, by defining an appropriate representer. And, in section 2.2.3, derivative constraints were implicity incorporated into the cost-function by formulating the approach in a *Sobolev Space* - which also permits an exact solution subject to the definition of the optimal kernel function.

However, what we would like is to incorporate such constraints without changing the model definition or kernel function, but instead by directly applying derivative constraints on the model directly - in a manner as consistent as possible with the RKHS approach of section 2.2.2.

In this section, we will formulate such an approach. By necessity, an approximation is required in the definition of the model. However, as will be discussed, it is easy to understand why such an approximation is necessary, and once understood, it is easy to formulate a corresponding solution.

### 3.4.1 Penalising Derivatives of Functions in the RKHS

Using (3.12), we can directly penalise functional properties by evaluating the Hilbert space norm over specified functional derivatives - in a similar fashion to (2.29). Hence, a regularisation over $\| f \|_{\mathcal{H}}$ is replaced by a constraint ensuring the model most adhering to a particular structural constraint is estimated. We nominally term this approach the *direct* approach, as constraints are evaluated in $\mathcal{H}$ :

$$\begin{aligned} \mathcal{J}_{\text{DREG}} : (f) &= \|\mathbf{y} - \mathbf{f}(\mathbf{x})\|_2^2 + \lambda \| \mathcal{D}^{\mathbf{m}} f \|_{\mathcal{H}}^2 \\ &= \sum_{i=1}^{\mathrm{N}} (y_i - \langle f, k_{x_i} \rangle_{\mathcal{H}})^2 + \lambda \langle \mathcal{D}^{\mathbf{m}} f, \mathcal{D}^{\mathbf{m}} f \rangle_{\mathcal{H}}. \end{aligned} \tag{3.19}$$

### 3.4.2 Determining the Optimal Model Configuration

For a model as per (2.24), the model parameters can be obtained by solving the following linear equation :

$$\left( \mathbf{K}^\top\mathbf{K} + \lambda\, \mathcal{D}^{(\mathbf{m},\mathbf{m})}\, \mathbf{K} \right) \alpha_{\mathcal{D}} = \mathbf{K}^\top\mathbf{y}. \tag{3.20}$$

Unlike in (2.31), the term $\mathbf{K}^\top$ cannot be factored out, hence care should be taken when solving for $\alpha$.

In this formulation, additional constraints on the model can be incorporated into $\mathcal{J}_{\text{DREG}}$ as desired, by adding derivative kernels into the solution ($\lambda\,\mathcal{D}^{(\mathbf{m},\mathbf{m})}\,\mathbf{K} \to \lambda_1\,\mathcal{D}^{(\mathbf{m_1},\mathbf{m_1})}\,\mathbf{K} + \lambda_2\,\mathcal{D}^{(\mathbf{m_2},\mathbf{m_2})}\,\mathbf{K}\ldots$). Similarly, constraints can be broken into directional constraints if desired in the same manner ($\lambda\,\mathcal{D}^{(\mathbf{m},\mathbf{m})}\,\mathbf{K} \to \lambda_1\,\partial_{x_1}^{(m_1,m_1)}\,\mathbf{K} + \lambda_2\,\partial_{x_2}^{(m_2,m_2)}\,\mathbf{K}\ldots$). In neither case does this affect either the definition of $\mathbf{K}$ or $\mathcal{F}$.

However, the representer $\mathcal{F}_{\text{RLS}}$ is suboptimal for the above problem. Whilst it can (and will) be used nonetheless, in the following section we will discuss why this is so, and how this can be dealt with.

### 3.4.3 A Representer for the Direct Approach

In [Schölkopf et al., 2001] it was shown that for certain cost-functions, *the reproducing property* (2.4) can be reduced from an infinite expansion to a finite sum over the observations. One of the criteria for this is that the regularisation term be a *monotically increasing* function on the norm of $f$. Clearly, monotonicity cannot be guaranteed for any arbitrary functional derivative $\|\mathcal{D}^{\mathbf{m}} f\|_{\mathcal{H}}$, and hence the representer theorem of (2.24) does not hold for the *direct approach* of $\mathcal{J}_{\text{DREG}}$ (3.19).

#### The Suboptimality of the Representer

This can be seen in different ways. Firstly, analysis of the representer term shows that the component of $f$ spanning the region of $\mathcal{H}$ not spanned by the observations is no longer irrelevant, unlike in (2.25):

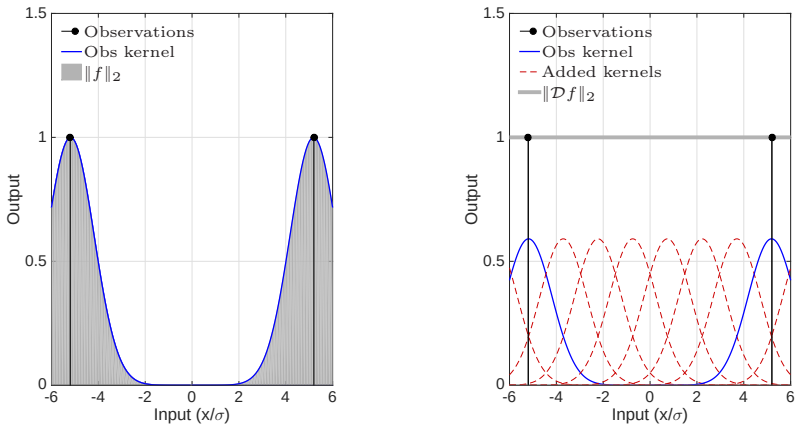
$$\begin{aligned}\|\mathcal{D}^{\mathbf{m}} f\|_{\mathcal{H}}^2 &= \langle \mathcal{D}^{\mathbf{m}} f, \mathcal{D}^{\mathbf{m}} f\rangle_{\mathcal{H}} \\ &= \langle \mathcal{D}^{\mathbf{m}} f_{\|} + \mathcal{D}^{\mathbf{m}} f_{\perp}, \mathcal{D}^{\mathbf{m}} f_{\|} + \mathcal{D}^{\mathbf{m}} f_{\perp}\rangle_{\mathcal{H}} \\ &= \langle \mathcal{D}^{\mathbf{m}} f_{\|}, \mathcal{D}^{\mathbf{m}} f_{\|}\rangle_{\mathcal{H}} + \langle \mathcal{D}^{\mathbf{m}} f_{\perp}, \mathcal{D}^{\mathbf{m}} f_{\perp}\rangle_{\mathcal{H}} + 2\langle \mathcal{D}^{\mathbf{m}} f_{\|}, \mathcal{D}^{\mathbf{m}} f_{\perp}\rangle_{\mathcal{H}}\end{aligned} \tag{3.21}$$

where $\langle \mathcal{D}^{\mathbf{m}} f_{\|}, \mathcal{D}^{\mathbf{m}} f_{\perp}\rangle_{\mathcal{H}}$ is possibly nonzero, and cannot be guaranteed to be positive. Therefore :

$$\begin{aligned}&\textit{if} \\ &\qquad \langle \mathcal{D}^{\mathbf{m}} f_{\|}, \mathcal{D}^{\mathbf{m}} f_{\perp}\rangle_{\mathcal{H}} < -\frac{1}{2}\langle \mathcal{D}^{\mathbf{m}} f_{\perp}, \mathcal{D}^{\mathbf{m}} f_{\perp}\rangle_{\mathcal{H}} \\ &\textit{then} \\ &\qquad \|\mathcal{D}^{\mathbf{m}} f_{\|}\|_{\mathcal{H}}^2 > \|\mathcal{D}^{\mathbf{m}} f\|_{\mathcal{H}}^2\end{aligned} \tag{3.22}$$

and hence $f(\mathbf{x}) = \sum_{i=1}^{\text{N}} \alpha_i\, k_{\mathbf{x}_i}(\mathbf{x})$ cannot be guaranteed to be the optimal representer of $\mathcal{J}_{\text{DREG}}$ (3.19).

We can also analyse this in a more intuitive fashion. To illustrate why this is the case, consider the example of Figure 3.8, with $\mathcal{X} \subset \mathbb{R}$ and $\mathcal{D}^{\mathbf{m}} = \mathcal{D}$ (i.e. minimisation of the gradient of $f$). It is considered that two measurements ($\mathcal{Z}_2 = \{(x_1, 1), (x_2, 1)\}$) of an unknown function $f_{\text{o}} : \mathcal{X} \to \mathbb{R}$ are observed.




(a) $\mathcal{J}_{\text{RLS}}$ : N = 2 (b) $\mathcal{J}_{\text{DREG}}$ : (N, M) = (2, 8)

FIGURE 3.8 – Re-approaching optimality using added kernels.

Here, the reasoning of equations (3.21) and (3.22) is illustrated for the case of the Gaussian RBF kernel. This extends directly to any RBF kernel, but not to any arbitrary kernel.

In Figure 3.8a, the function is reconstructed using the approach of $\mathcal{J}_{\text{RLS}}$, using the representer $\mathcal{F}_{\text{RLS}}$, for a fixed kernel width - which is assumed to be small relative to the spacing of the observations. Placing kernels in this way at the observations allows the error of the model to be minimised - no more kernels are required in $\mathcal{X}$ to do so, nor would adding kernels improve the results. Similarly, adding kernels along $\mathcal{X}$ would only increase $\|\hat{f}\|_{\mathcal{H}}$ - hence $\mathcal{F}_{\text{RLS}}$ is optimal in this case.

By constrast, in Figure 3.8b the objective of the regularisation term is to minimise $\|\mathcal{D}\hat{f}\|_{\mathcal{H}}$. Again, kernels at the observations are required to minimise the error, but unlike for $\mathcal{J}_{\text{RLS}}$, adding kernels along $\mathcal{X}$ would facilitate the minimisation of $\|\mathcal{D}\hat{f}\|_{\mathcal{H}}$, rather than hinder it. Hence, without also constraining the behaviour of the kernel function, $\mathcal{F}_{\text{RLS}}$ cannot be the optimal representer of $\mathcal{J}_{\text{DREG}}$.

### Resolving the Suboptimality

Without constraining the kernel function, an optimal representer for $\mathcal{J}_{\text{DREG}}$ cannot be defined - as an infinite number of kernels would be required to fully minimise $\|\mathcal{D}\hat{f}\|_{\mathcal{H}}$.

However, it can be ensured that $\|\mathcal{D}\hat{f}\|_{\mathcal{H}}$ is *small* if the density of kernels in $\mathcal{X}$ is sufficiently large. If observations are densely distributed in $\mathcal{X}$, a flexible kernel can be chosen with respect to the data (as will be described shortly).

But if the observations are not sufficiently dense, flexibility in $\mathbf{K}$ can be ensured by adding kernels along $\mathcal{X}$ - effectively approximating the infinite representation of $f(\mathbf{x})$ in (2.4).

Accordingly, we propose an *extended representer* of $f$ for $\mathcal{J}_{\text{DREG}}$, based on the true observations and added grid points.

$$\mathcal{F}_{\text{EXT}} : f(\mathbf{x}) = \sum_{i=1}^{\text{N}} \alpha_i k_{\mathbf{x}_i}(\mathbf{x}) + \sum_{j=1}^{\text{M}} \alpha_j^* k_{\mathbf{x}_j^*}(\mathbf{x}), \tag{3.23}$$

where $\mathbf{x_j}^* \in \mathbb{R}^{n_x}, j \in [1 \ldots \mathrm{M}]$ are points arbitrarily defined, e.g. as a uniform gridding of $\mathcal{X}$.

The use of an extended representer allows the user greater control over the definition of $\mathcal{X}$, as the model definition is no longer completely dependent on the data. However, whilst this can be attractive in smaller problems (as will be shown in section 4.2), in larger problems the definition of a grid could be cumbersome and computationally expensive. Hence, as an alternative, we also propose a heuristic method for kernel selection - which can be applied to either the extended representer or the usual representer.

This method also serves as a way of evaluating the suboptimality of the chosen representer for $\mathcal{J}_{\mathrm{DREG}}$. This does require a constraint on the kernel function: for example, it applies a minimum bound on the width of the Gaussian RBF kernel. However, this is a soft constraint (in the sense that it can be applied as seen fit), and in practice allows the selection of a smaller kernel (and therefore a more flexible model class) than would otherwise be possible using the normal RKHS approach.

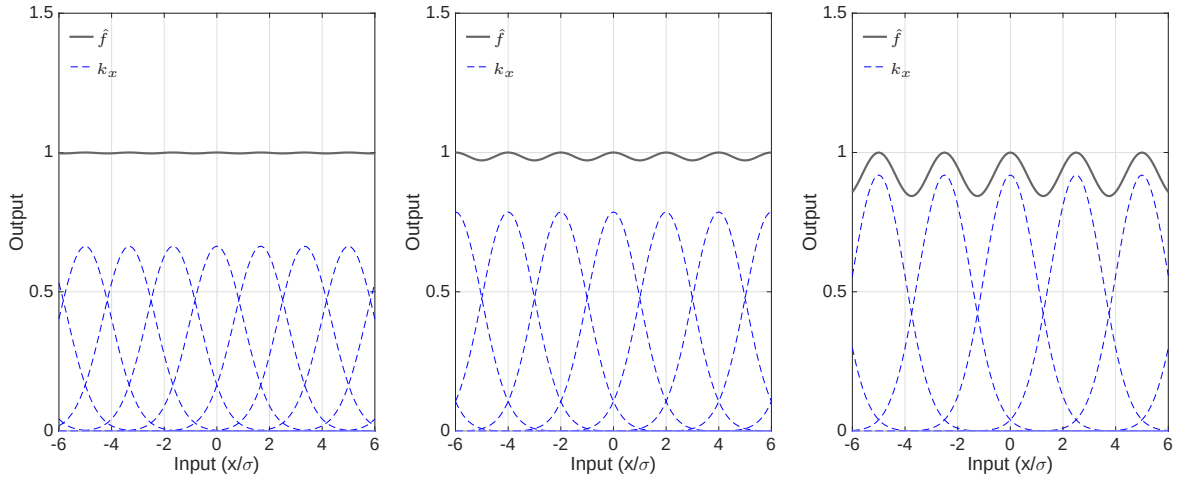
Again, this analysis will be applied in the case of the Gaussian kernel: however it can be repeated for any RBF kernel.

### Evaluating the Suboptimality

The extent to which the representer is suboptimal depends on the density of kernels in $\mathcal{X}$ - i.e. the relation between the width of kernels and their spacing:

$$\rho_k = \frac{\sigma}{\Delta_x}, \quad \Delta_x = \max(x_i - x_{i-1}), \tag{3.24}$$

for adjacent $x_i - x_{i-1}$. Figure 3.9 illustrates this relation: as the density of kernels reduces, the ability of the model to reconstruct a constant function also reduces. It is this capability that we will now try to evaluate.




(a) $\rho_k = 0.6$ (b) $\rho_k = 0.5$ (c) $\rho_k = 0.4$

FIGURE 3.9 – The effect of the kernel density in $\mathcal{X}$ on $\hat{f}(x)$.

To determine a suitable value of $\rho_k$, we will now introduce a *smoothness tolerance parameter* $\epsilon$ is now introduced :

$$\epsilon_{\hat{f}} = 100 \times \left\{ 1 - \frac{\|\hat{f}\|_\infty}{C} \right\} \%. \tag{3.25}$$

$\epsilon$ is a user-defined tolerance, quantifying the maximum relative difference between $f$ and $C \in \mathbb{R}$. Whilst $\epsilon$ changes with the definition of the kernel, and what constitutes a suitable value of $\epsilon$ will depend on the application in question, the curve itself is not data-sensitive in that it relates to the maximum spacing.

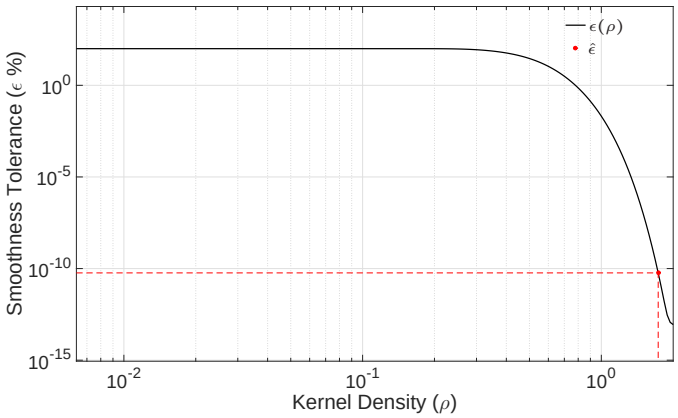



FIGURE 3.10 – Selecting an appropriate kernel using $\epsilon$

Figure 3.10 shows a numerical evaluation of $\epsilon$ over a range of kernel density values. As can be seen, the cost of the approximation in the representer is a negligible for larger values of $\rho_k$ - with $\epsilon < 1\%$ at $\rho_k > 1$.

In practice, a reasonable choice of $1.7 \leq \rho_k \leq 2$ - which ensures $\epsilon < 10^{-10}\%$. In this way, an appropriate value for the kernel can be chosen *a priori* - allowing a flexible model class definition with smoothness properties tuned through the kernel function. Alternatively, the kernel width can always be included in the hyperparameter optimisation problem - as would usually be done.

**Remark 1:** If a uniform smoothness tolerance is acceptable, the expression of (3.24) can be generalised to higher dimensions:

$$\sigma_i \geq \sqrt{n_x}\,(\,\rho_k\,\Delta_{x_i}\,)\,, \tag{3.26}$$

for $i = 1, \ldots, n_x$. Note, $\Delta_{x_i}$ is the maximum spacing along each axis of the input space. Whilst this is harder to evaluate in higher dimensional problems, $\Delta_{x_i}$ can be approximated by averaging out N over $\mathcal{X}$, or by using geometric approaches such as *Delauney Triangulation* (which is feasible for $n_x \leq 4$).

**Remark 2:** This example considers the estimation of a constant as the limit case for $\mathcal{J}_{\text{DREG}}$ in the case of $\mathcal{R}(f) = \lambda \|\mathcal{D} f\|^2_{\mathcal{H}}$. However, this case actually extends to all derivative operators: if a constant function can be reasonably approximated, any derivative constraint can be applied using an RBF kernel.

### 3.4.4 Formulating a Bias Function

In a similar manner to the Sobolev Space case, a bias function can also be included in the model definition for $\mathcal{J}_{\text{DREG}}$. Whilst it may not be strictly necessary for all kernels in all problems, in many cases it is advantageous as it allows full application of the regularisation constraint. The reasoning for this is similar to the Sobolev case, where the boundary conditions of the Green's function problem ensure that the function is bounded to 0 at the limits of $\mathcal{X}$.

For example, in the case of a constraint minimising the gradient of $f$:

$$\lim_{\lambda \to \infty} \lambda \| \mathcal{D} f \|_{\mathcal{H}}^2 = 0 \quad \Rightarrow \quad f(\mathbf{x}) = c, \ \forall x \in \mathcal{X}, \ c \in \mathbb{R}. \tag{3.27}$$

We can see that for kernels that are bounded to 0 at some point in $\mathcal{X}$ (e.g. $x = 0$ for polynomials kernels when $\gamma_1 = 0$) the only function that is smooth everywhere in $\mathcal{X}$ is the zero function $f(\mathbf{x}) = 0, \ \forall x \in \mathcal{X}$. The same effect is present in the Gaussian RBF kernel, which decreases away from its centre

$$\lim_{r \to \infty} \mathrm{K}(\mathbf{x}_i, \mathbf{x}_j) = \exp(-r^2/\sigma^2) = 0, \quad r = \| \mathbf{x}_j - \mathbf{x}_i \|_2. \tag{3.28}$$

For many classes of kernel functions, a derivative regularisation constraint will enforce a zero function in the limit case in a similar manner to the functional norm regularisation, but we can expect that the trajectory of this constraint will differ. To ensure a non-zero limit case, it is then necessary to include a function exclusively spanning the null space of the derivative operator. Hence the analogy with the *bias function* used for spline kernels.

If a *linear, parametric* bias function is included, in a similar fashion to (2.30), the solution for the model parameters in (3.20) becomes:

$$\begin{bmatrix} \mathbf{X}^\top \mathbf{X} & \mathbf{X}^\top \mathbf{K} \\ \mathbf{K}^\top \mathbf{X} & \mathbf{K}^\top \mathbf{K} + \lambda \mathcal{D}^{(\mathbf{m},\mathbf{m})} \mathbf{K} \end{bmatrix} \begin{bmatrix} \theta \\ \alpha \end{bmatrix} = \begin{bmatrix} \mathbf{X}^\top \\ \mathbf{K}^\top \end{bmatrix} \mathbf{y}. \tag{3.29}$$

The difference between the above solution and (2.32) can be seen as a result of the suboptimality of the representer.

## 3.5 A Comparative Example

To compare the two methods presented in this chapter with those presented in the previous chapter, the example of section 2.4 will be recalled. The exact same data is used, with the experiment carried out in the same manner as before: the optimal model configuration is determined, and afterwards the models are re-estimated over a range of hyperparameters, to show how properties can be controlled in each case.

### 3.5.1 Experimental Procedure

To estimate models using the indirect and direct methods, the following procedure was used.

1. $\mathcal{M}_{\text{DMIN}}$: here, the indirect method of section 3.3 was used with a Gaussian RBF kernel, and a constraint minimising evaluations of the second derivative of the model. The model

parameters were estimated using the expression in (3.17) and the derivative kernels given in section 3.2.1.

The model configuration was determined by optimising all three hyperparameters $(\sigma, \lambda_1, \lambda_2)$ using cross-validation in the manner discussed in section 2.4.2.

2. $\mathcal{M}_{\text{DREG}}$: here, the direct method of section 3.4 was used, also with a Gaussian RBF kernel. As in $\mathcal{M}_{\text{CSK}}$, a linear bias component was included in the model definition, with the model parameters determined using (3.20).

   The width of the kernel was fixed *a priori* to a suitable value, determined by calculating the maximum spacing of observations in $\mathcal{X}$, and letting $\rho_k = 2$, as described in section 3.4.3. Hence, only optimisation over the regularisation hyperparameter is considered - in a similar fashion to $\mathcal{M}_{\text{CSK}}$ - which was performed using cross-validation, as for the other models.

The performance of the optimal models was evaluated as before, using the expressions given in (2.34), with the results displayed in Table 3.1 for all four methods.

### 3.5.2 Optimal Results

| | FIT % | MBIAS | MSDEV | MRMSE | $\sigma$ | $\log_{10}\lambda$ |
|---|---|---|---|---|---|---|
| $\mathcal{M}_{\text{GRBF}}$ | 91.42 | .0345 | $3.38 \times 10^{-4}$ | .0300 | 0.31 | 0.27 |
| $\mathcal{M}_{\text{CSK}}$ | 88.92 | .0348 | $1.27 \times 10^{-4}$ | .0336 | N/A | −2.57 |
| $\mathcal{M}_{\text{DMIN}}$ | 88.16 | .0393 | $7.09 \times 10^{-5}$ | .0391 | 0.22 | −2.75, −0.04 |
| $\mathcal{M}_{\text{DREG}}$ | 90.04 | .0334 | $2.28 \times 10^{-4}$ | .0304 | 0.10 | 0.87 |

TABLE 3.1 – Summarised results of the optimised models in Figure 3.11.

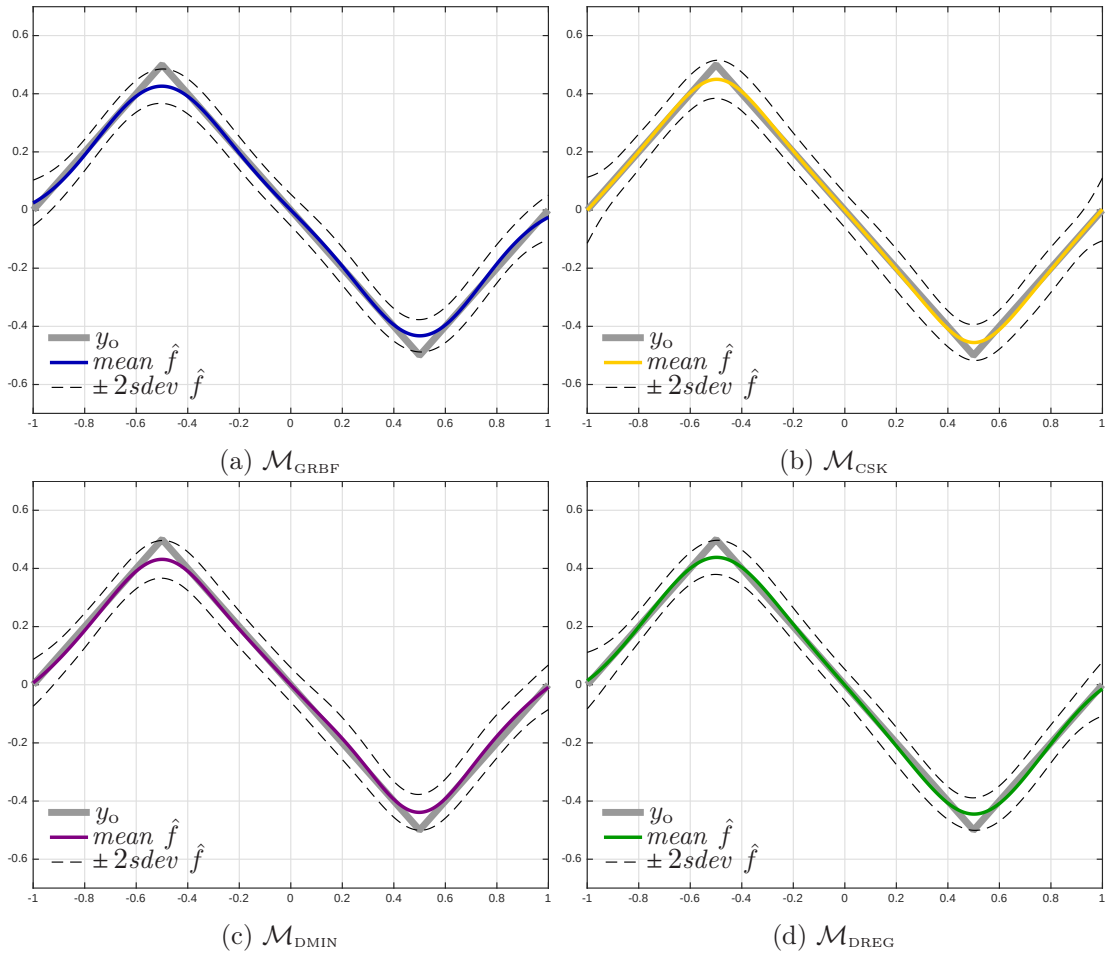
Table 3.1 shows that again, all methods perform similarly well. Interestingly, the addition of a derivative penalty in $\mathcal{M}_{\text{DMIN}}$ actually introduces a slight bias into the model with respect to $\mathcal{M}_{\text{GRBF}}$, however in Figures 3.11a and 3.11c, the two models are almost indistinguishable. Similarly, $\mathcal{M}_{\text{CSK}}$ and $\mathcal{M}_{\text{DREG}}$ are also extremely similar, which shows that the *direct* method of penalising derivatives in an RKHS can - for a proper choice of kernel - achieve the same performance as the optimal Sobolev kernel approach.

This is really interesting because now we have two methods (the direct and indirect approaches) with which properties can be controlled in a *soft* manner, as in a Sobolev space method. However, as the penalties are formed in an RKHS, there is still freedom in the choice of kernel function.

### 3.5.3 Tuning the Model Smoothness

| | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| $\log_{10}\lambda_{\text{DMIN}}$ | −8 | 2.1 | 4.6 | 5.8 | 8 |
| $\log_{10}\lambda_{\text{DREG}}$ | −3 | −1.14 | 1.8 | 2.4 | 5.1 |

TABLE 3.2 – Hyperparameter values for Figure 3.12.




(a) $\mathcal{M}_{\text{GRBF}}$ (b) $\mathcal{M}_{\text{CSK}}$

(c) $\mathcal{M}_{\text{DMIN}}$ (d) $\mathcal{M}_{\text{DREG}}$

FIGURE 3.11 – Optimal results.

As in section 2.4.4, the models $\mathcal{M}_{\text{DMIN}}$ and $\mathcal{M}_{\text{DREG}}$ were re-estimated over a range of hyperparameter values. In the case of $\mathcal{M}_{\text{DMIN}}$, $\lambda_1$ was varied whilst for $\mathcal{M}_{\text{DREG}}$ $\lambda$ was varied. Sample results are plotted in Figure 3.12, using the hyperparameter values given in Table 3.2.

In the case of $\mathcal{M}_{\text{DMIN}}$, rather than using the optimised kernel width ($\sigma = 0.22$), the value used for the direct method was taken ($\sigma = 0.1$), to better illustrate how the derivative constraint acts on the smoothness properties of the model, by reducing the influence of the kernel function.

As can be seen in Figure 3.12, the results are almost identical to those presented in section 2.4.4. In particular, between the smoothing spline estimate and the direct approach (centre and right) there are almost no visible differences in how the properties of the model change. The sole exception is the first figure, in which the 'low bias' model is achieved for much lower variance using $\mathcal{M}_{\text{DREG}}$ than for $\mathcal{M}_{\text{CSK}}$. This is due to the effect of the kernel which, whilst set to an *a priori* small value, still placed some constraints on the model class - and hence allows an accurate estimate with reasonable reliability.

Whilst this example is very simple, it is nonetheless useful as aid for understanding how the four approaches work. In a one-dimensional static problem, the model selection is reduced to a smoothness detection problem, hence the effects of each approach can be compared by examining how the model smoothness is controlled.

Importantly, we now not only have a framework for applying soft constraints on the model, by one that performs as well as the theoretically exact approach.

## 3.6 Summary

| | *Method* | $\mathcal{C}(f)$ | $\mathcal{H}$ | + + + | − − − |
|---|---|---|---|---|---|
| I. | RKHS method | $\|f\|^2_{\mathcal{H}}$ | RKHS | exact sol<br>free $\mathbf{K}$ | |
| II. | Spline smoothing | $\int_{\mathcal{X}} \left(\mathcal{D}f\right)^2 \mathrm{d}x$ | *Sobolev* | exact sol | $\mathcal{D}, \mathbf{K}$ not free<br>partial $\mathcal{D}$ ? |
| III. | Indirect method | $\sum_{k=1}^{\mathrm{M}} (\mathcal{D}f(x_k))^2$ | RKHS | exact sol<br>free $\mathcal{D}$ and $\mathbf{K}$ | evals<br>params |
| IV. | Direct method | $\|\mathcal{D}f\|^2_{\mathcal{H}}$ | RKHS | global<br>free $\mathcal{D}$ and $\mathbf{K}$ | suboptimal<br>approx sol |

Table 3.3 – Summary of the methods presented in Chapters 2 and 3.

In this chapter, several new ideas have been introduced. Firstly, it was shown that derivatives are well-formulated in the RKHS - and that they can be computed easily using the *derivative reproducing property* (3.11).

Using these results, two methods of incorporating derivatives into the optimisation scheme were presented, the direct and indirect methods. The two methods are distinct, but both allow the penalisation of derivatives in an RKHS, without need to formulate the problem in a Sobolev Space.

Both methods were formulated thoroughly, with consideration of a representer and how this relates to a kernel function highlighted in each case as an important concern. The approximation made by both methods depends on proper selection of the kernel function, hence the kernel function is still loosely constrained. In practice, this presents little obstacle, with the user free either to define an *a priori* choice of kernel function, with respect eiter to the data or to a gridding, or to include the kernel in the optimisation - as would usually be done.

With this fact properly understood, the user is now free to incorporate derivative constraints as desired. Whilst in certain cases, it wil also be possible to formulate the equivalent Sobolev Space problem, as a rule-of-thumb differentiation is easier than integration - and determining a derivative operator is easier than solving a Green's function integral equation.

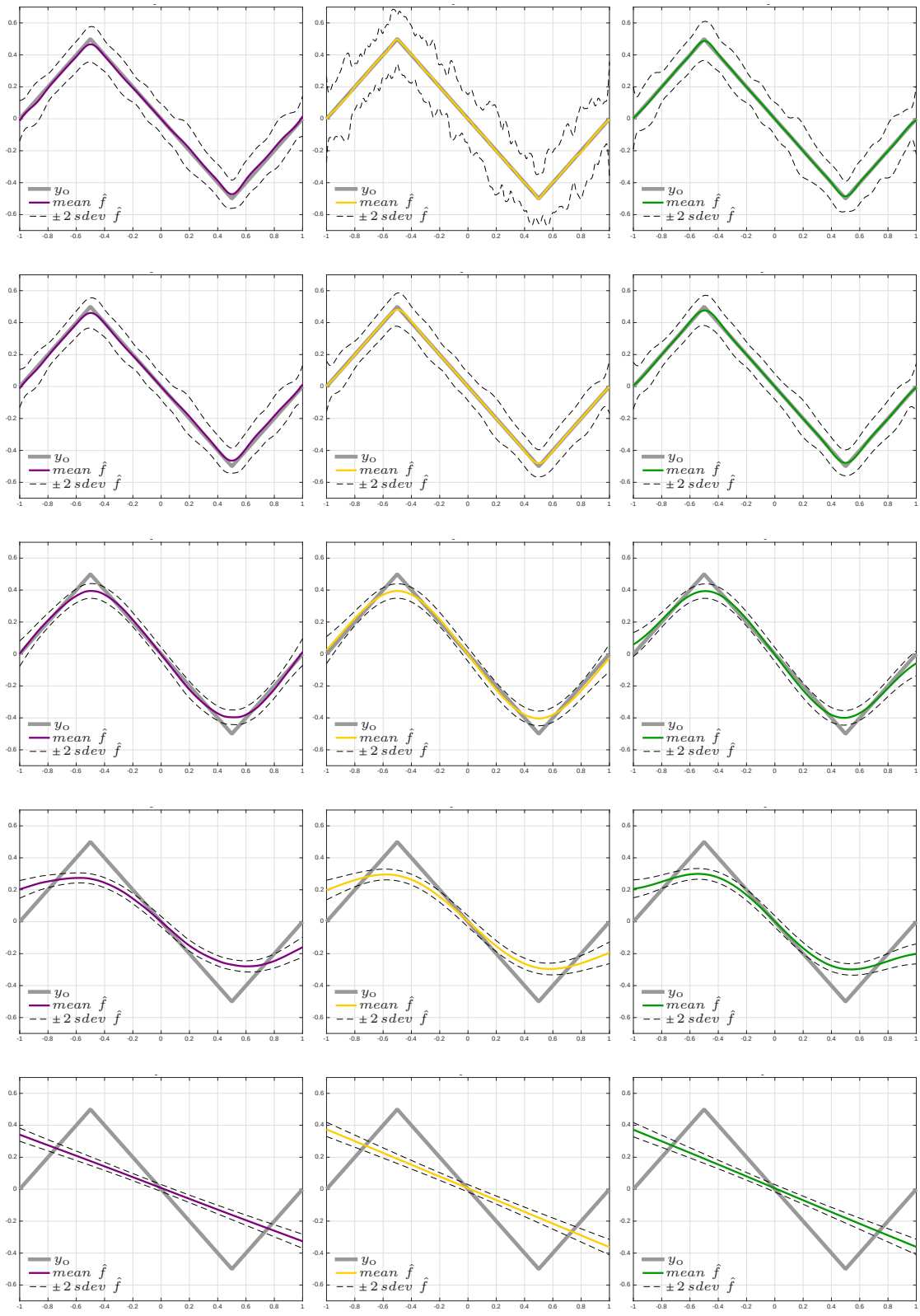



FIGURE 3.12 – Comparing the smoothing effect of the *indirect method* (left) and *direct method* (right) with the smoothing splines (centre).

# 4

# Identification Case Studies

*In this chapter, the methods developed in the previous chapters (in particular, the 'direct' method) will be applied to several simulated case studies, with the intention of illustrating how the combination of hard and soft model constraints can be useful in practical identification problems.*

## 4.1 Introduction

So far, four distinct approaches to kernel-based nonlinear identification in the RKHS have been formulated and discussed. For convenience, henceforth the following abbreviations will be used to refer to each approach:

1. F-REG : the functional norm regularisation approach to RKHS identification, discussed in section 2.3.3. In this approach, tuning of the model properties is achieved through the choice of kernel, with the regularisation used to ensure numerical stability.
2. SPLINES : similar to the F-REG approach, but with the function defined in a *Sobolev Space*. Allows tuning of the model properties through a regularisation term, with the choice of kernel function implicit in the problem definition, as discussed in section 2.2.3.
3. D-MIN : i.e. the *indirect* approach to derivative penalisation in the RKHS (section 3.3). Here, tuning of the model can be achieved both through the kernel function *and* through a regularisation term, by penalising evaluations of functional derivatives in $\ell_2$.
4. D-REG : i.e. the *direct* approach to derivative penalisation in the RKHS (section 3.4). Also allows tuning both through the kernel function and through a regularisation term, by penalising derivatives of the function in an RKHS using a regularisation term (similarly to SPLINES ).

The first three approaches are all formulated in a theoretically exact manner, whereas the final approach relies on an approximation to the optimal solution. This can be understood either by analysis of the representer (as in the F-REG case), or through the kernel selection (as in the SPLINES case). However, as shown in chapter 3, whilst the formulation of D-MIN is theoretically exact, proper consideration should still be given to the specification of the kernel function in a similar fashion to in D-REG .

At this point, we can draw several preliminary conclusions, notably :

- In the RKHS, most methods focus on proper estimation of the kernel function: enforcing a *hard* constraint on the model properties, through the model class.
- However, as demonstrated in chapter 2, in certain situations *hard* and *soft* optimisation of the model properties can achieve almost identical results. This gives the user new possibilities in terms of how model properties can be controlled, and what types of models can be estimated in practice.
- In the case of SPLINES the choice of kernel function is implicit in the definition of $\|f\|_{\mathcal{S}_k}$, hence determining the optimal kernel for different nonlinear problems is not necessarily straightforward.
- However, the approaches presented in chapter 3 can achieve good results with respect to the *optimal* approach based on a Sobolev space formulation, and can be readily applied to different problems through the choice of kernel function and the definition of a derivative operator.

In this chapter we will present three simulation examples, each of which designed to move away from the notion of *equivalence*. Instead, we will highlight cases in which the application of soft model class constraints confers some type of advantage with respect to the classical RKHS approach.

The first example will be a simple one-dimensional static function, possessing both smooth and nonsmooth elements. It will be shown that a soft optimisation approach outperforms F-REG in this scenario due to several key features :

- The ability to deconstrain the model class, reducing the extent to which any *a priori* constraint on the smoothness of the model must be made.
- And the ability to vary the smoothness of the model over the input space as required.

The second example will show how a soft optimisation approach can allow for structral approximation in dynamical model structures. Specifically, we will consider the estimation of LTI-like LPV models. It will be shown that, whilst the *optimal* configuration of F-REG and D-REG are very similar, the use of a derivative constraint proffers the user a way to trade-off between the accuracy of the model and its applicability.

The final example will show how D-REG can be combined with the F-REG approach in practical scenarios for complexity tuning and variance minimisation. This will focus on the application of structural penalties in nonlinear dynamical models, which is - to the best of the author's knowledge - an otherwise unstudied area in nonlinear identification. And, unlike in the first and second examples, in this case finding the exact solution to SPLINES using a Sobolev space approach is non-trivial, and hence currently uninvestigated.

This chapter will focus on the application of smooth, global constraints on the model. As such, we will focus on the use of D-REG as oppposed to D-MIN . As shown in Chapter 3, for such problems equivalent results can be achieved using either approach. The scope of D-MIN in nonlinear identification will be touched upon again at the end of this chapter.

## 4.2 Structural Detection Using Smoothness Constraints

The first example aims to show how a smoothness-enforcing approach can be advantageous in applications such as structural detection, i.e. situations in which the estimated model should

not only function as an accurate predictor of the data - but also provide the user with useful indicators regarding the behaviour of the system.

The advantage of D-REG with respect to F-REG in this case stems from its ability to deconstrain the model class, by choosing an *a priori* flexible kernel function and instead placing a soft constraint through the regularisation. This reduces the extent to which the model is forced to *average out* the smoothness of the system, reducing the amount of information lost in the estimation process.

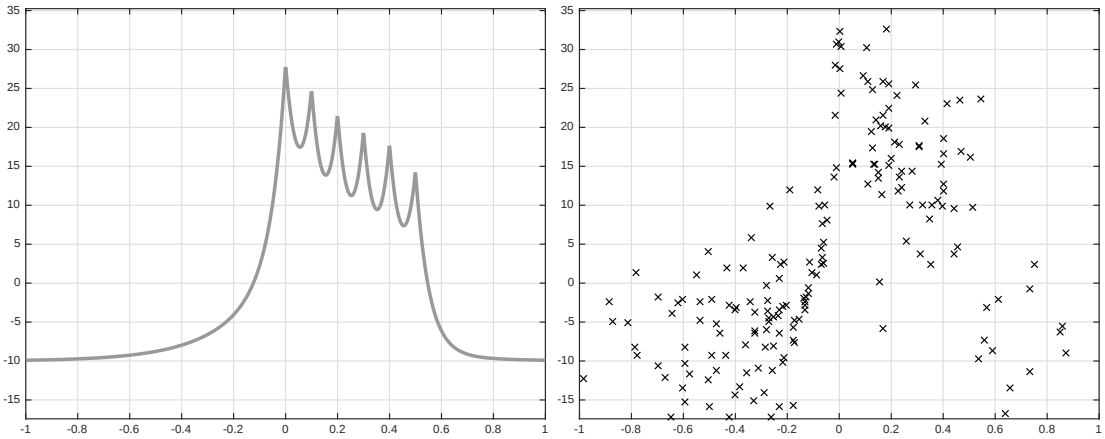
### 4.2.1 The Data-Generating System

The simulated system is designed to reflect the types of problems often seen in biological applications. A one-dimensional, static nonlinear function is considered, with both smooth and nonsmooth components :

$$\mathcal{S}_o : \; f_o(x) = 15e^{-5|x|} + 20\sum_{i=0}^{5} e^{-20|x-0.1i|} - 10, \quad x \in \mathbb{R}. \tag{4.1}$$

The nonsmooth behaviour here could relate to the activation of neurons or genes in a regulatory network, with the smooth component perhaps relating to a nonlinear flow.

We consider that relatively few observations are available (N = 100), normally distributed across the input space ($x_i \sim \mathcal{N}(0, 0.5)$). And, these observations are highly noisy, with measurement corrupted at the output by white Gaussian noise of SNR = 5dB, where SNR = $20\log\left(\sigma_{f_o}/\sigma_e\right)$. This reflects both the high cost of performing experiments in biological applications, and the difficulty in ensuring the accuracy of any obtained measurements.



(a) Noise-free function (b) Noisy observations

FIGURE 4.1 – The data-generating system of (4.1).

The data-generating system is visible in Figure 4.1a, with the observed data visible in Figure 4.1b. Due to the noise-level and the complexity of the nonlinear function, visually determining the structure of 4.1 from Figure 4.1b is very challenging.

### 4.2.2 Identification Procedure

Two distinct approaches to the modelling problem were considered :

- $\mathcal{M}_1$: F-REG - with a Gaussian RBF kernel. The hyperparameters, $\sigma$ and $\lambda$, were optimised by cross-validation as described below.
- $\mathcal{M}_2$: D1-REG - i.e. D-REG with a regularisation term constraining the first derivative (i.e. the gradient) of the model. Here, $\lambda$ was optimised by cross-validation, as described below. And a Gaussian RBF kernel was used in this case as well. But rather than optimising the kernel width, $\mathrm{P}_{\max} = 10^3$ additional grid points were added to the model (as described in section 3.4.3), and $\sigma$ was determined with respect to the data for $\rho_k = 2$.

To estimate the models in each case, a two-step procedure was used:

1. The optimal model configuration was determined using a validation set (or cross-validation) approach, as described in [James et al., 2014], i.e. models were estimated using the training data and evaluated against a validation dataset generated under identical conditions. The optimal model configuration in each case is taken to be the one minimising the error (MSE) of the model with respect to the noisy validation data.
2. Following determination of the optimal model configuration, Monte-Carlo trials were run to test the performance of each algorithm. In this case $n_{mc} = 10^3$ trials were run, in which the noise-free training data was subject to different noise realisations.

In each case, the performance of the models was evaluated against a gridded dataset (with $n_{grid} = 10^3$), with the exception of the FIT value - which was evaluated against the noise-free validation data. The corresponding results are displayed in Table 2.1, calculated using the following expressions :

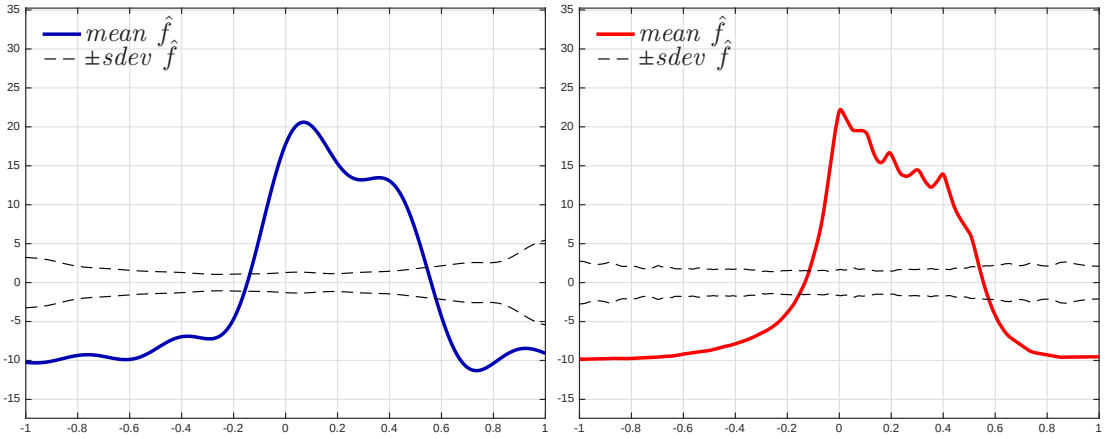
$$\begin{aligned}
\text{FIT} &= 100 \cdot \left(1 - \frac{\left\| y - \hat{f} \right\|_2^2}{\left\| y - \bar{y} \right\|_2^2}\right) \\
\text{MBIAS} &= \text{mean}_{i=1}^{n_{\text{GRID}}} \left\| y_{\text{o},i} - \text{mean}_{j=1}^{n_{\text{MC}}} \{\hat{f}_j(x_i)\} \right\|_1 \\
\text{MSDEV} &= \text{mean}_{i=1}^{n_{\text{GRID}}} \left\| \sigma_{\hat{f}}(x_i) \right\|_2 \\
\text{MRMSE} &= \text{mean}_{i=1}^{n_{\text{MC}}} \left\| y - \hat{f}_i \right\|_2 .
\end{aligned} \tag{4.2}$$

### 4.2.3 Results

From the plots of Figure 4.2, we can see that both models fail to properly model the spikes of (4.1). However, D1-REG - whilst unable to fully capture the spikes - is clearly able to distinguish between the smooth behaviour at the extremeties of $\mathcal{X}$ and the nonsmooth behaviour between $x = 0$ and $x = 0.5$.

Furthermore, the estimation along the smooth regions is also greatly improved, with the sharp descent and smooth profiles at $x < 0$ modelled with much greater accuracy than with the F-REG approach. These observations are supported by the results in Table 4.1, which shows a greatly reduced bias is achieved for equivalent variance.

Of course, in such an example, without sufficient observations the results will always depend on the particular experiment. Clearly, using a smoothness-enforcing regularisation does not allow us to estimate what is not present in the data - nor would we want to do that. But, by using a




(a) F-REG (b) D1-REG

FIGURE 4.2 – Unlike in the examples of the previous chapters, a clear difference between the estimated models is visible.

| | FIT | MBIAS | MSDEV | MRMSE | $\sigma$ | $\log_{10}\lambda$ |
|---|---|---|---|---|---|---|
| F-REG | 77.41% | 4.12 | 1.90 | 2.90 | 0.360 | -2.20 |
| D1-REG | 84.13% | 1.93 | 1.90 | 2.36 | 0.005 | 1.59 |

TABLE 4.1 – Summarised results of the optimised models in Figure 4.2.

flexible kernel function and a soft constraint on the smoothness, we are at least able to minimise the amount of information lost.

Furthermore, it is worth noting that this is achieved *without* making any prior assumptions regarding the presence or lack thereof of either smooth or nonsmooth phenomena in the system. It simply takes maximum advantage of the flexibility afforded to it, and lets the data inform the modelling process as much as possible.

## 4.3 Controlling Smoothness in Dynamical Models

In the previous section, the ability of a smoothing regulariser to deconstrain the model class in a simple, static nonlinear problem was considered. In this section, the same method will be applied to a simulated dynamical problem, more reflective of the type of systems usually seen in system identification.

As an example, we will consider the case of *linear parameter-varying* (LPV) systems. As discussed in chapter 1, the estimation of accurate LPV models is an area of considerable interest in system identification - as such models both well-approximate many real-world nonlinear systems and are well understood from a control perspective.

Firstly, the F-REG RKHS method of chapter 2 will be formulated for the LPV case. Next, it will be shown how this formulation can be straightforwardly extended to incorporate the D-REG method. These two approaches will then be applied to a simulation example, to show how both methods can achieve similar results. Finally, the two approaches will be combined to show how

hard constraints can be incorporated into the model through the kernel, and the regularisation term can be used to act on structural properties of the model.

### 4.3.1 Modelling LPV Systems Using RKHS Methods

To the best of the author's knowledge, at present a thorough treatment of the LPV identification problem from an RKHS perspective has not been considered. Nonetheless, by analogy with other kernel-based solutions in the literature (e.g. [Tóth et al., 2011, Laurain et al., 2012, Piga and Tóth, 2013, Golabi et al., 2014, Darwish et al., 2015]), it will be shown that a solution for the problem of (1.8) can be presented in the RKHS.

We will start by recalling the definition of an LPV model from chapter 1 :

$$\mathcal{M}_{lpv}: \quad f_{lpv}(x_k, p_k) = \sum_{i=1}^{n_x} f_{nl_i}(p_k)\, x_{i,k}, \qquad f_i : \mathbb{P} \to \mathbb{R} \tag{4.3}$$

As can be seen, an LPV model maintains a linear relationship between the regressor variables and the output variables, but this relationship depends upon a nonlinear *scheduling function*. In a parametric context, the user explictly defines the nonlinear functions $f_{nl_i}(p_k)$. However in a nonparametric formulation, the linear and nonlinear components are both defined in terms of kernel functions. Hence, by forming kernels for each component function, we can reconstruct an LPV model.

Furthermore, the linearity constraint on the input variable can be incorporated into the model simply by using a linear kernel (2.8). This follows from the theory of sums and products of reproducing kernels [Aronszajn, 1950], i.e. that sums and products of kernels are also kernels, defining a corresponding RKHS composed of sums and products of the individual RKHSs :

$$\begin{aligned} \mathrm{K}_{lpv}([\mathbf{x}, p], [\mathbf{x}^*, p^*]) &= \sum_{i=1}^{n_x} \mathrm{K}_{lpv_i}([x_i, p], [x_i^*, p^*]) \\ &= \sum_{i=1}^{n_x} x_i\, \mathrm{K}_{nl_i}(p, p^*)\, x_i^*. \end{aligned} \tag{4.4}$$

A model is defined using combinations of kernels, weighted across $\mathcal{H}$ :

$$\begin{aligned} f(\mathbf{x}^*, p^*) &= \sum_{k=1}^{\mathrm{N}} \alpha_k\, \mathrm{K}_{lpv}([\mathbf{x_k}, p_k], [\mathbf{x}^*, p^*]) \\ &= \sum_{i=1}^{n_x} \sum_{k=1}^{\mathrm{N}} \alpha_k x_{i,k} \mathrm{K}_{nl_i}(p_k, p^*) x_i^*. \end{aligned} \tag{4.5}$$

Now, subject to the definition of a suitable $n_p$-dimensional kernel describing the underlying scheduling nonlinearities $f_{nl_i}$, $\mathrm{K}_{nl_i}(p_j, p_k)$, the identification of a model can be carried out in exactly the same manner as before.

As expected, this is consistent with the solutions developed in other kernel frameworks, for example as in the *least-squares support vector machine* approach presented in [Tóth et al., 2011] :

$$(\mathbf{\Omega} + \lambda \mathbf{I})\, \alpha = \mathbf{y}. \tag{4.6}$$

It can be seen that $\mathbf{\Omega}$ and $\mathbf{K}$ in (2.31) are equivalent. In [Tóth et al., 2011], $f$ is defined as per (4.5) and $\mathbf{\Omega_{j,k}} = \sum_{i=1}^{n_x} x_{i,j} \mathrm{K}_{nl_i}(p_j, p_k) x_{i,k}$.

Similarly, note from [Tóth et al., 2011] that the scheduling functions can be computed according to :

$$f_{nl_i}(p^*) = \sum_{k=1}^{N} \alpha_k x_{i,k} \mathrm{K}_{nl_i}(p_k, p^*), \text{ for } i = 1 \ldots n_x. \tag{4.7}$$

In an RKHS setting, evaluating $f_{nl_i}(p^*)$ corresponds the evaluation of $f(\mathbf{x}^*, p^*)$ in (4.5) at $\{\mathbf{x}^*\}_{j=1}^{n_x} = \delta_{i,j}$, where $\delta_{i,j}$ is a *Dirac delta function*. It should be noted that isolating individual components of a model in the RKHS is not always possible. It is possible in this case because of the particular structure of the LPV kernel function.

### 4.3.2 Penalising Derivatives of LPV Models

Just as in the previous section, where RKHS methods were applied to the LPV modelling problem, derivative penalties can be straightforwardly applied to RKHS methods for LPV identification.

As noted in section 3.2.2, derivatives are well-formulated in the RKHS provided $\mathcal{H}$ is sufficiently differentiable in the relevant direction. In an LPV setting, this means that although derivatives of $f$ can be taken with respect to any scheduling variables, they *cannot* be taken with respect to the input/regressor variables.

With this in mind, we define a smoothness operator for LPV models in the RKHS as follows :

$$\mathcal{D}_p^m f(\mathbf{x_k}, p_k) = [\partial_{p_1}^m(\mathbf{x_k}, p_k) \cdots \partial_{p_{n_p}}^m f(\mathbf{x_k}, p_k)]^\top, \tag{4.8}$$

where $\partial_{p_i}^m\{\cdot\} = \frac{\partial^m}{\partial {p_i}^m}\{\cdot\}$ for $i = 1, \ldots n_p$, i.e. the $m^{\text{th}}$-order partial derivative of $f$ with respect to the $i^{\text{th}}$ dimension of $p$.

The corresponding norm of the derivative can be evaluated in the same manner as before, by determining the relevant derivative kernel function. For example, in the simplest case ($n_p = 1$, $m \in \mathbb{Z}^+$) :

$$\begin{aligned} \|\mathcal{D}_p^m f\|_{\mathcal{H}}^2 &= \sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \, \mathcal{D}_p^{(m,m)} \mathrm{K}_{lpv}([x_i, p_i], [x_j, p_j]) \, \alpha_j \\ &= \sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{k=1}^{n_x} \alpha_i \, x_{i,k} \, \frac{\partial^{2m} \mathrm{K}_{nl}([x_i, p_i], [x_j, p_j])}{\partial p_i^m \, \partial p_j^m} \, x_{j,k} \, \alpha_j. \end{aligned} \tag{4.9}$$

Therefore, computation of the derivative of an LPV kernel is equivalent to the computation of an LPV kernel, with $\mathrm{K}_{nl}$ replaced by $\mathcal{D}\mathrm{K}_{nl}$. And, with this result the solution of (3.20) can be applied as usual.

At this point, it is necessary to make several remarks prior to proceeding further.

**Remark 1:** It is important to note that in the LPV framework $\partial_{p_i}^m f \neq \partial_{p_i}^m f_i$, i.e. the above formulation does not explicitly place separate constraints on the different scheduling functions. It may be possible to place such constraints, particularly by noting that :

$$\begin{aligned} \|\partial_{p_i}^m f\|_{\mathcal{H}}^2 &= \sum_{i=1}^{n_x} \|\partial_{p_i}^m f_i x_i\|_{\mathcal{H}}^2 \\ &= \sum_{i=1}^{n_x} \|\partial_{p_i}^m f_i\|_{\mathcal{H}}^2 \|x_i\|_{\mathcal{H}}^2, \end{aligned} \tag{4.10}$$

if $p_i \nsim x_i$. This is not investigated here and it will be shown that nonetheless, adequate control over the smoothness of $f$ can be obtained using D-REG . Alternatively, use of D-MIN would allow direct penalisation of the scheduling functions, in a similar manner to [Piga and Tóth, 2013]. However as discussed in [Zhou, 2008], this would require an additional set of parameters for each scheduling function contrained.

**Remark 2:** As in the previous chapter, the definition of a suitable representer for D-REG is again an important point. In example of the following section, the kernel hyperparameter will be determined with respect to the observations, in a similar fashion to the example of section 4.2. Alternatively, the kernel hyperparameter can be included in the optimisation problem as would otherwise be the case for F-REG .

The question of how to efficiently define a suitable gridding for an LPV model remains an open question at this point.

**Remark 3:** Similarly, a discussion concerning the formulation of a suitable bias function will be neglected for the time being. However, as will be shown presently, adequate results can be obtained without its consideration.

### 4.3.3 Simulation Example

To compare the performance of the two formulated methods, a simulation example will now be presented.

#### The Data-Generating System

The following LPV-ARX system will be considered:

$$\mathcal{S}_{lpv} : y_k = \sum_{i=1}^{n_x} f_{\mathrm{o},i}(p_k) x_{i,k} + e_{\mathrm{o},k}, \tag{4.11}$$

for $n_a = 1$, $n_b = 1$, $n_u = 1\,(\therefore n_x = 3)$ and $n_p = 1$. The coefficient functions of (4.11) are given by the following equations :

$$\begin{aligned} f_{\mathrm{o},1}(p_k) &= \begin{cases} 0.5\,(1 - |p_k|), & 0.5 < p_k < -0.5 \\ -p_k, & -0.5 \leq p_k \leq 0.5 \end{cases} \\ f_{\mathrm{o},1}(p_k) &= 0.7\mathrm{sinc}(7\pi^2 p_k) \\ f_{\mathrm{o},3}(p_k) &= -0.5(1 - \exp\{1 - |p_k|\}). \end{aligned} \tag{4.12}$$

The coefficient functions of (4.12) each possess different smoothness characteristics - which both adhere to and violate the smoothness assumptions of the models generated in different ways. This provides a clear way of testing the effect of each model hypothesis on the resulting estimation.

101 experiments were run with 2N = 2000 measurements of the system excited by a uniformly distributed input signal and scheduling signal : $u_k \sim U(-1, 1)$, $p_k \sim U(-1, 1)$, $k = 1 \ldots 2\mathrm{N}$. The observations were corrupted by a white noise disturbance $e_{\mathrm{o},k} \sim N(0, \sigma_e^2)$, with $\sigma_e = 0.015$, such that $\mathrm{SNR} = 20 \log\left(\sigma_{y_\mathrm{o}}/\sigma_e\right) \approx 5\mathrm{dB}$.

| | $\sigma$ | $\log_{10}(\lambda)$ | Fit (%) |
|---|---|---|---|
| F-REG | 0.100 | 0.00 | 76.11 |
| D1-REG | 0.035 | −2.13 | 76.44 |
| D2-REG | 0.035 | −4.97 | 76.37 |

TABLE 4.2 – Summarised results.

| F-REG | $f_1$ | $f_2$ | $f_3$ |
|---|---|---|---|
| MBIAS | 0.027 | 0.016 | 0.019 |
| MSDEV | 0.055 | 0.035 | 0.037 |
| MRMSE | 0.62 | 0.41 | 0.42 |

TABLE 4.3 – F-REG results.

| D1-REG | $f_1$ | $f_2$ | $f_3$ |
|---|---|---|---|
| MBIAS | 0.010 | 0.021 | 0.008 |
| MSDEV | 0.046 | 0.031 | 0.033 |
| MRMSE | 0.47 | 0.45 | 0.35 |

TABLE 4.4 – D1-REG results.

| D2-REG | $f_1$ | $f_2$ | $f_3$ |
|---|---|---|---|
| MBIAS | 0.009 | 0.025 | 0.007 |
| MSDEV | 0.052 | 0.032 | 0.033 |
| MRMSE | 0.52 | 0.49 | 0.35 |

TABLE 4.5 – D2-REG results.

Each dataset was split into two equal subsets of N = 1000 points, corresponding to an estimation set and a validation set in each case. The first experiment was used to determine the optimal model configuration, and the remaining $n_{\text{MC}} = 100$ experiments used for Monte-Carlo simulations.

**Experimental Procedure**

Three models were identified, according to the following procedures.

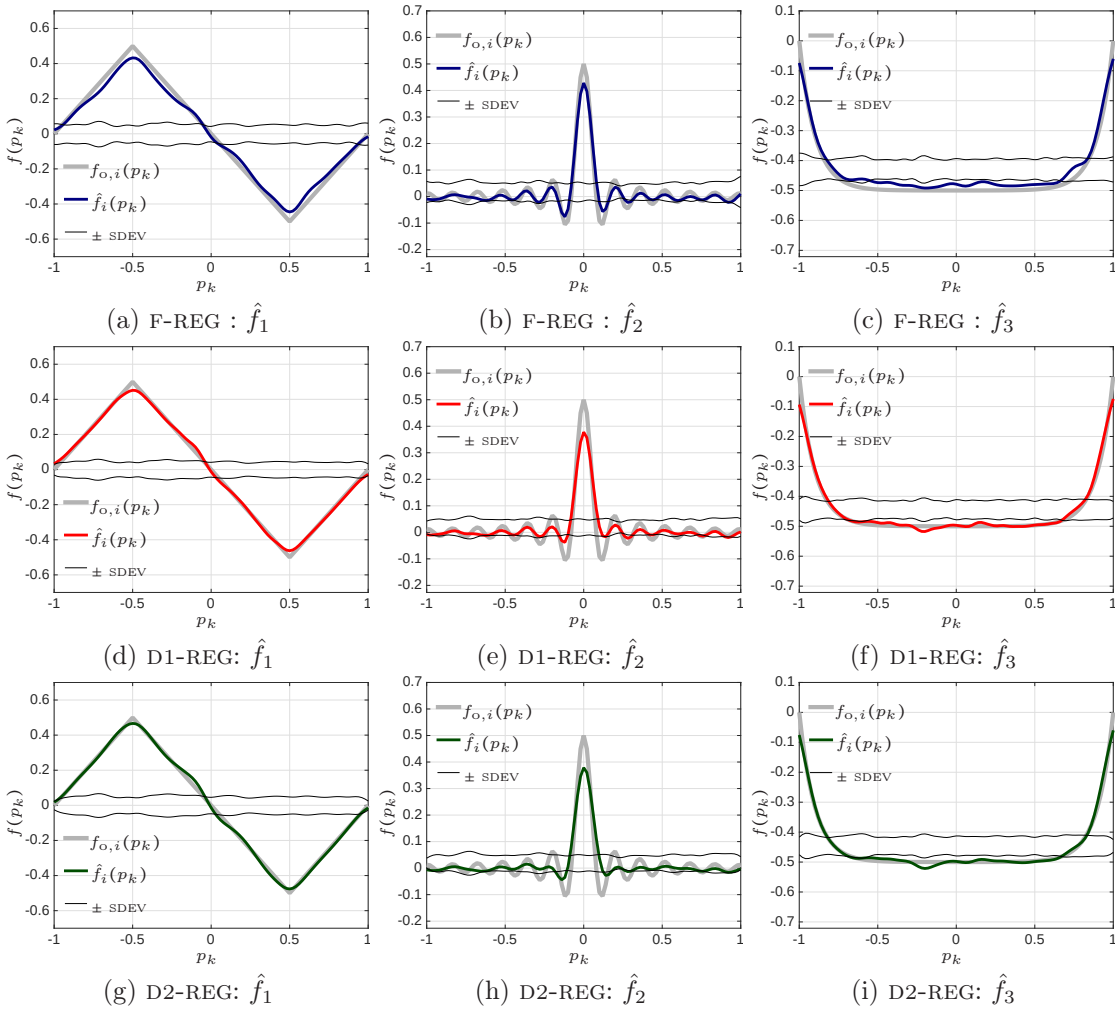
- F-REG , with a Gaussian RBF kernel used to describe the nonlinear scheduling dependency. A uniform kernel width was used for each coefficient function, resulting in two hyperparameters for optimisation ($\sigma$ and $\lambda$). The model hyperparameters were determined by cross-validation.
- D1-REG , i.e. D-REG with the first derivative of $f$ with respect to $p$ constrained. Again, a Gaussian kernel of uniform width was used to describe the scheduling dependencies.
- D2-REG , i.e. D-REG with the second derivative of $f$ with respect to $p$ constrained, and a Gaussian kernel of uniform width describing the scheduling dependencies.

In the case of the latter two approaches, the regularisation hyperparameter $\lambda$ was determined by cross-validation, whilst the kernel width $\sigma$ was determined with respect to the spacing of the observations of $p$ in $\mathbb{P}$, ensuring a density of $\rho_k = 2$ in each case. Note that, unlike in section 4.2, no additional grid points were added to the representer in either case.

Following determination of the optimal model configuration (Table 4.2), Monte-Carlo trials were performed. The performance of the estimated models was evaluated, by computing the coefficient functions in each case (according to (4.7)), using the criteria described in (2.34). These results are summarised in Tables 4.3 - 4.5 and Figure 4.3.

**Results**

From Figure 4.3 and Tables 4.3 - 4.5, it is clear that the optimised models perform similarly in each case - showing how the control of smoothness can be equivalently managed through the regularisation if desired. The differences in the results can be seen to reflect the differing characteristics of the coefficient functions :




(a) F-REG : $\hat{f}_1$ (b) F-REG : $\hat{f}_2$ (c) F-REG : $\hat{f}_3$

(d) D1-REG: $\hat{f}_1$ (e) D1-REG: $\hat{f}_2$ (f) D1-REG: $\hat{f}_3$

(g) D2-REG: $\hat{f}_1$ (h) D2-REG: $\hat{f}_2$ (i) D2-REG: $\hat{f}_3$

FIGURE 4.3 – Results of Monte-Carlo simulations on models of $\mathcal{S}_{lpv}$ for optimised hyperparameter values

— F-REG is better able to estimate $f_{o,2}$ - as its comparatively higher energy in the first and second derivatives makes it very costly for the approaches of D1-REG and D2-REG , resulting in the introduction of a bias (visible in Figures 4.3e and 4.3h).
— Similarly, F-REG is significantly more biased in the estimation of $f_{o,1}$ and $f_{o,3}$ - as these functions better reflect the model assumptions of D1-REG and D2-REG .

Between the estimates of D1-REG and D2-REG , the difference is minimal, with D1-REG showing a marginally lower variance and D2-REG a marginally lower bias. Again, this can be seen to reflect the different penalties each approach introduces into the optimisation scheme.

### 4.3.4 Tuning the Model Properties

So far, it has been shown that equivalent control of the smoothness can be achieved either through the kernel function or through a regularisation term. But, in the framework of D-REG , nothing prevents the user from combining these two constraints. To illustrate this point, D1-REG and D2-REG were re-estimated with $\sigma = 0.1$ (i.e. the optimal kernel width of F-REG in the previous

section), with the results displayed in Table 4.6. Now, as the smoothness is controlled through the kernel, the smoothness constraints can become *structural* constraints on the model.

| | $\sigma$ | $\log_{10}(\lambda)$ | Fit (%) |
|---|---|---|---|
| D1-REG | 0.1 | −2.48 | 76.33 |
| D2-REG | 0.1 | −5.68 | 76.24 |

TABLE 4.6 – Re-tuned model configuration.

For instance, ensuring a constant gradient along $\mathbb{P}$ enforces an LTI-like approximation to $\mathcal{S}_{lpv}$. Whilst applying a constraint on the second derivative ensures a linear-like scheduling dependency.

This offers the user a new degree of freedom in the identification problem. Optimal results can be estimated as usual, and the complexity of the model additionally tuned through a regularisation term as desired. Now however, acting on the variance of the model also has a physical signficance in terms of its properties. That is, in this situation, a lower variance model is also a physically simpler model.

This becomes even more interesting when we consider how this can be combined with a bias function. In a strict mathematical sense, a bias function corresponds to the null space of the derivative operator. However, as illustrated in the previous section, good results can be obtained in practice without consideration of the bias function - which in turn, allows the user some freedom with respect to how such a function can be designed in practice.

Some of the most popular approaches to parametric LPV identification in the literature focus on the estimation of locally LTI and piecewise-linear LPV models. These approximations are very useful from a control perspective, but limited if the behaviour of the true system is far from the model set. Hence, combination of an *ideal* parametric LPV structure and a corresponding nonparametric LPV structure could allow for reduced error with respect to parametric approaches, whilst additionally allow the degree of approximation in the system to be freely controlled through a regularisation hyperparameter.

## 4.4 Complexity Tuning Using Structural Penalties in Nonlinear Models

In the previous section, the notion of derivative penalties as structural constraints was touched upon, however this was still in the context of smoothness penalties. This section will go one step further, and investigate the scope of regularisation terms solely acting on interactions between variables, i.e. penalties that act exclusively on the structure of $f$ and not on its smoothness.

To the best of the authors' knowledge, this is an entirely novel approach in nonlinear system identification. Hence, this section will begin with a slightly more general discussion on separable models, before formulating a soft regularisation based approach. Finally, two simulation examples will be presented. The first of which will simply aim to illustrate the effect of a structural penalty on the model properties, whilst the second will focus on how such constraints can be incorporated into the identification of nonlinear dynamical systems in practice.

### 4.4.1 Evaluating Structural Properties Using Derivatives

Up to this point, the types of derivative operators considered have all been derivatives or partial derivatives of a single variable, used to enforce some type of smoothness constraint. Such constraints can also be readily extended to multivariate problems. However, in section 3.2.3 the idea of combining derivatives to form *interaction kernels* was introduced. In fact, these interaction kernels can be linked to a deeper notion of structure in nonlinear models.

As an example, we consider a function $f : \mathbb{R}^2 \to \mathbb{R}$ of two independent variables $x_1$ and $x_2$. It will be assumed that $f$ lies in an RKHS $\mathcal{H}$, and is smooth, such that derivatives are well-defined up to a certain order as required.

The interaction kernel of Figure 3.6a used an operator $\partial_{x_1}\partial_{x_2}\{\cdot\}$. Evaluating $f$ with the same operator applied allows us to draw several conclusions :

**Case 1: Nonseparable**

$$\left\| \frac{\partial^2 f(x_1, x_2)}{\partial x_1 \, \partial x_2} \right\|_{\mathcal{H}} \neq 0. \tag{4.13}$$

In this case, the function is nonseparable, that is it depends on both $x_1$ and $x_2$, and this relationship cannot be further simplified without making further assumptions.

**Case 2: Additive Separable**

$$\left\| \frac{\partial^2 f(x_1, x_2)}{\partial x_1 \, \partial x_2} \right\|_{\mathcal{H}} = 0. \tag{4.14}$$

In this case, the function is *additively separable*, i.e. $f$ can be decomposed into two independent functions $g_1 : \mathbb{R} \to \mathbb{R} \in \mathcal{H}_1$ and $g_1 : \mathbb{R} \to \mathbb{R} \in \mathcal{H}_2$ such that $\mathcal{H} = \mathcal{H}_1 + \mathcal{H}_2$ :

$$\left\| \frac{\partial^2 f(x_1, x_2)}{\partial x_1 \, \partial x_2} \right\|_{\mathcal{H}} = 0 \Rightarrow f(x_1, x_2) = g_1(x_1) + g_2(x_2). \tag{4.15}$$

Note that no statement has been made here about either $g_1$ or $g_2$, other than that they are both non-trivial and differentiable.

**Case 3: Quasi-separable**

$$\left\| \frac{\partial^2 f(x_1, x_2)}{\partial x_1 \, \partial x_2} \right\|_{\mathcal{H}} < \epsilon, \tag{4.16}$$

where $\epsilon$ represents some *small* value, e.g. $0 < \epsilon << \| f \|_{\mathcal{H}}$. Although in this case, the function cannot be exclusively written as an additively separable function, the presence of interactions in the function should be small. For example, if $f$ is expressed as a sum of additive and non-additive components $f(x_1, x_2) = g_1(x_1) + g_2(x_2) + h(x_1, x_2)$, for $h : \mathbb{R}^2 \to \mathbb{R}$, it is reasonable to expect that

$$\left\| \frac{\partial^2 f(x_1, x_2)}{\partial x_1 \, \partial x_2} \right\|_{\mathcal{H}} < \epsilon \Rightarrow \|h\|_{\mathcal{H}} << \|g_1\|_{\mathcal{H}_1} + \|g_2\|_{\mathcal{H}_2}. \tag{4.17}$$

Here, no statement is made regarding *how* the function is 'quasi-separable', or what this means. It could be that the function is additive separable everywhere, with the exception of a small region

(as for example, in the case of switching functions), or perhaps that the function is globally nonseparable, but $\frac{\partial^2 f(x_1,x_2)}{\partial x_1 \partial x_2} \approx 0$ over all values $x_1$ and $x_2$.

**Remark:** As a side note, here the use of derivatives to evaluate additive separability is considered. *Multiplicatively* separable functions, i.e. functions that can be decomposed into products of functions $f(x_1, x_2) = g_1(x_1) \times g_2(x_2)$, are not. Multiplicative separability cannot be evaluated without also placing assumptions on the differentiability of the component functions $g_1$ and $g_2$. For example,

$$\left\| \frac{\partial^2 g_1(x_1) g_2(x_2)}{\partial x_1 \partial x_2} \right\|_{\mathcal{H}} = 0 \Rightarrow \frac{\partial g_1}{\partial x_1} \text{ and/or } \frac{\partial g_2}{\partial x_2} = 0. \tag{4.18}$$

However, this could be of interest in model structures where such constraints are inherently made, such as LPV or Hammerstein models.

Furthermore, whilst additive separability can be straightforwardly incorporated into the kernel function (by summing kernels, as discussed in section 2.2.2), multiplicative separability cannot [Poggio and Girosi, 1990]. Hence, the investigation of derivative operators evaluating such properties could be an interesting course of future research.

### 4.4.2 Penalising Separability Using Functional Derivatives

As in section 3.2.2, these properties can be evaluated using kernels :

$$\begin{aligned} \frac{\partial^2 f(x_1,x_2)}{\partial x_1 \partial x_2} &= \langle f, \partial_{x_1}\partial_{x_2} k_{\mathbf{x}} \rangle_{\mathcal{H}} \\ &= \sum_i \alpha_i \, \partial_{x_1}\partial_{x_2} k_{\mathbf{x}_i}(\mathbf{x}), \end{aligned} \tag{4.19}$$

$$\begin{aligned} \left\| \frac{\partial^2 f(x_1,x_2)}{\partial x_1 \partial x_2} \right\|_{\mathcal{H}}^2 &= \partial_{x_1}^{(1,1)} \partial_{x_2}^{(1,1)} \mathbf{K} \\ &= \sum_i \sum_j \alpha_i \alpha_j \frac{\partial^4 \mathrm{K}(\mathbf{x}_i, \mathbf{x}_j)}{\partial x_{1,i} \partial x_{1,j} \partial x_{2,i} \partial x_{2,j}}. \end{aligned} \tag{4.20}$$

Therefore, a constraint on the separability of $f$ can be directly incorporated into the optimisation scheme using D-REG . This approach is nominally termed DX-REG , with DX denoting a penalty on the cross-derivatives of $f$:

$$\text{DX-REG}: \quad \mathcal{J}(f) = \sum_{k=1}^{\mathrm{N}} \left( y_k - f(\mathbf{x_k}) \right)^2 + \lambda \| \partial_{x_1}\partial_{x_2} f \|_{\mathcal{H}}^2. \tag{4.21}$$

In a similar fashion to the smoothness-enforcing approaches discussed up to this point, a *bias* function can be added the model. In this case, a reasonable choice of bias function is an additive component, such that the model definition becomes

$$\begin{aligned} f(\mathbf{x}) &= f_{\text{ADD}}(\mathbf{x}) + f_{\text{NL}}(\mathbf{x}) \\ &= \sum_{i=1}^{\mathrm{N}} \sum_{j=1}^{n_x} \alpha_i \, k_{x_{i,j}}(x_j) + \sum_{i=1}^{\mathrm{N}} \beta_i \prod_{j=1}^{n_x} k_{x_{i,j}}(x_j) \\ &= \mathbf{K}_{\text{ADD}} \alpha + \mathbf{K}_{\text{NL}} \beta. \end{aligned} \tag{4.22}$$

As this additive component is also a nonparametric function, it is strongly advisable to consider the addition of a constraint on $f_{\text{ADD}}$ in the optimisation scheme, e.g. $\mathcal{R}(f_{\text{ADD}}) = \lambda \| f_{\text{ADD}} \|^2_{\mathcal{H}}$. In which case, the model parameters $\alpha, \beta \in \mathbb{R}^{\text{N}}$ of (4.22) can be determined by solving the following expression:

$$\begin{bmatrix} \mathbf{K}_{\text{ADD}}^{\top}\mathbf{K}_{\text{ADD}} + \lambda_1 \mathbf{K}_{\text{ADD}} & \mathbf{K}_{\text{ADD}}^{\top}\mathbf{K}_{\text{NL}} \\ \mathbf{K}_{\text{NL}}^{\top}\mathbf{K}_{\text{ADD}} & \mathbf{K}_{\text{NL}}^{\top}\mathbf{K}_{\text{NL}} + \lambda_2 \partial_{x_1}^{(1,1)} \partial_{x_2}^{(1,1)} \mathbf{K}_{\text{NL}} \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \mathbf{K}_{\text{ADD}}^{\top} \\ \mathbf{K}_{\text{NL}}^{\top} \end{bmatrix} \mathbf{y}. \tag{4.23}$$

Note that, whilst the kernels $\mathbf{K}_{\text{ADD}}, \mathbf{K}_{\text{NL}} \in \mathbb{R}^{\text{N}\times\text{N}}$ are inherently different from each other due to their structure, in practice the same one-dimensional kernels should be used in each case to ensure $\mathcal{H}_{\text{ADD}} \subset \mathcal{H}_{\text{NL}}$. For example, if a Gaussian kernel is used, the kernel widths of each kernel should be the same for each variable $x_i$. However, the kernel widths could still be different for different $x_i$.

**Remark:** To the best of the authors' knowledge, a comparable approach *exclusively* penalising the separability of the model is difficult to achieve in Sobolev framework. Discussion in the literature regarding additive and interaction splines focuses on the explicit definition of functions modelling additive behaviour and interactions between variables (see for example Chapter 10 of [Wahba, 1990]). This is nonetheless a very interesting approach, but distinct from the DX-REG method considered here.

### 4.4.3 Simulation Example 1

Before proceeding further, we will now present the first of the two simulation examples considered in this section. Here, it is aimed simply to show *how* the separability penalty formulated in the previous section acts on the model properties.

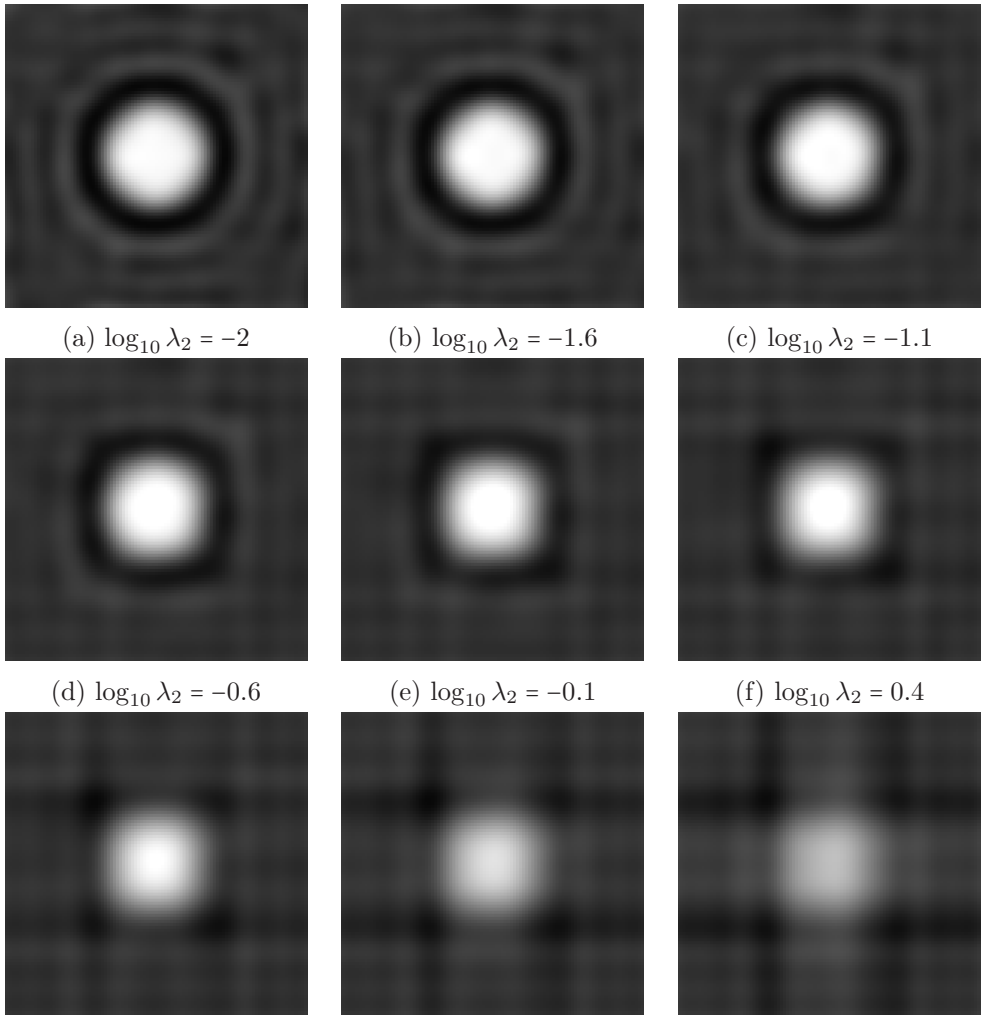
To this end, a smooth, nonseparable function is considered :

$$\mathcal{S}_{\text{o}} : \begin{cases} y_k & = f_{\text{o}}(u_{1,k}, u_{2,k}) + e_{\text{o},k}, \\ f_{\text{o}}(x_1, x_2) & = \operatorname{sinc}\big([x_1^2 + x_2^2]\big). \end{cases} \tag{4.24}$$

This bears close analogy with the use of a nonsmooth function to analyse the effect of a smoothing regularisation term in sections 2.4 and 3.5. N = 500 observations of the system were measured using inputs normally distributed across $\mathcal{X} = \mathbb{R}^2$, $u_{1,k}, u_{2,k} \sim \mathcal{N}(0, 0.35)$. DX-REG was trained using these measurements, with $\mathbf{K}_{\text{ADD}}$ and $\mathbf{K}_{\text{NL}}$ constructed using a Gaussian RBF kernel of width $\sigma = 0.2$ in each direction. The regularisation hyperparameter $\lambda_1$ was fixed *a priori* at 0.1, and $\lambda_2$ was varied over a range of values. 100 Monte-Carlo trials were performed with a noise level of SNR= 10dB in each case.

The estimated models were plotted over a two-dimensional grid between -1 and 1, with the mean results for increasing values of $\lambda_2$ displayed in Figure 4.4.

In Figure 4.4a, the estimated model is visible for a low value of $\lambda_2$, such that the impact of the separability constraint is negligible. As $\lambda_2$ is increased, the effect of the regularisation becomes increasingly apparent. In particular, Figures 4.4d - 4.4f show how the regularisation acts on the model away from $\mathbf{u} = (0, 0)$ by 'straightening out' the model. This straightening is a direct result of the reduced interactions in the model. As the observations are normally distributed in $\mathcal{X}$, the model is more inclined to approximate the system away from the centre of $\mathcal{X}$, where the fewest measurements are recorded.



(a) $\log_{10}\lambda_2 = -2$ (b) $\log_{10}\lambda_2 = -1.6$ (c) $\log_{10}\lambda_2 = -1.1$

(d) $\log_{10}\lambda_2 = -0.6$ (e) $\log_{10}\lambda_2 = -0.1$ (f) $\log_{10}\lambda_2 = 0.4$

(g) $\log_{10}\lambda_2 = 1.0$ (h) $\log_{10}\lambda_2 = 1.5$ (i) $\log_{10}\lambda_2 = 2.0$

FIGURE 4.4 – Applying penalities such as a *separability* penalty can allow for structural approximation in the model. Here the regularisation can be seen to act on the *shape* of $\hat{f}$.

By contrast, even as the regularisation is further increased (Figures 4.4g - 4.4i), the roundedness of the model close to $\mathbf{u} = (0, 0)$ is preserved.

This action on the *shape* of the model is a direct consequence of the penalty term used. By comparison, using the F-REG approach with the model of (4.22), it should be possible to obtain results similar to those visible in Figures 4.4a and 4.4i. But achieving something close to the intermediate results would be extremely difficult, as such an optimisation scheme has no way of continuously penalising structural properties such as separability without also acting on another property, such as smoothness.

### 4.4.4 Applying Structural Constraints to Practical Identification Problems

In the previous section, an example was used to show how a separability penalty acts on the model properties. Whilst a new tool for acting on the model is interesting in itself, it isn't really clear at this point how such a penalty could be used in practice. Whilst there could be many potential uses of such a penalty, e.g. as a way of gaining insight into the behaviour of a system, how can such penalties help in the estimation of better nonlinear models ?

One possible answer to the above question is simply that they offer new possibilities with respect to how the complexity of the model can be tuned. In the F-REG and SPLINES methods, the complexity of the model is tuned principally either through the 'minimality' of the model (i.e. by constraining its norm) or through its smoothness. Separability is a weaker constraint than either of these, in the sense that an additive nonlinear function is more complex than either a linear function or a zero function. Hence, enforcing separability can reduce the extent to which minimality or smoothness must be used as approximations in nonlinear models.

Whilst there are many ways in which such penalities could be formulated, here just one will be considered. The operator defined in (4.19) can be extended from $\mathcal{X} = \mathbb{R}^2$ to $\mathcal{X} = \mathbb{R}^{n_x}$ by constraining interactions between all variables in the model :

$$\mathcal{D}^{\mathrm{x}}(\cdot) = [\ \partial_{x_1}\partial_{x_2}(\cdot) \quad \partial_{x_1}\partial_{x_3}(\cdot) \quad \cdots \quad \partial_{x_{n_x-1}}\partial_{x_{n_x}}(\cdot)\ ]^\top. \tag{4.25}$$

Again, to express this using kernels we take the *derivative reproducing property* of (3.11) :

$$\begin{aligned}\mathcal{D}^{\mathrm{x}} f(\mathbf{x}) &= \langle f, \mathcal{D}^{\mathrm{x}} k_{\mathbf{x}} \rangle_{\mathcal{H}} \\ &= [\ \langle f, \partial_{x_1}\partial_{x_2} k_{\mathbf{x}}\rangle_{\mathcal{H}} \quad \langle f, \partial_{x_1}\partial_{x_3} k_{\mathbf{x}}\rangle_{\mathcal{H}} \quad \cdots \quad \langle f, \partial_{x_{n_x-1}}\partial_{x_{n_x}} k_{\mathbf{x}}\rangle_{\mathcal{H}}\ ]^\top.\end{aligned} \tag{4.26}$$

To formulate a norm, we evaluate the sum of the norms of individual interaction penalties.

$$\begin{aligned}\mathcal{D}^{(\mathrm{x},\mathrm{x})}\mathbf{K} &= \left\langle \mathcal{D}^{\mathrm{x}} k_{\mathbf{x}}, \mathcal{D}^{\mathrm{x}} k_{\mathbf{x}^*} \right\rangle_{\mathcal{H}} \\ &= \sum_{i=1}^{n_x-1} \sum_{j>i}^{n_x} \left\langle \partial_{x_i}\partial_{x_j} k_{\mathbf{x}}, \partial_{x_i^*}\partial_{x_j^*} k_{\mathbf{x}^*} \right\rangle_{\mathcal{H}} \\ &= \sum_{i=1}^{n_x-1} \sum_{j>i}^{n_x} \partial_{x_i}^{(1,1)}\partial_{x_j}^{(1,1)}\, \mathbf{K}(\mathbf{x}, \mathbf{x}^*).\end{aligned} \tag{4.27}$$

Note that this is just one example of many types of penalties that can be formed. In fact, comparison with the *thin-plate splines* approach (see Chapter 2 of [Wahba, 1990]) shows that penalising interactions in this way is equivalent to taking only the cross-derivative penalties. The addition of a smoothing term could be used to formulate an equivalent penalty if desired, but this isn't considered here.

Additional constraints on the model can also be included as desired, for example on the norm $\|f_{\mathrm{NL}}(\mathbf{x})\|^2_{\mathcal{H}}$:

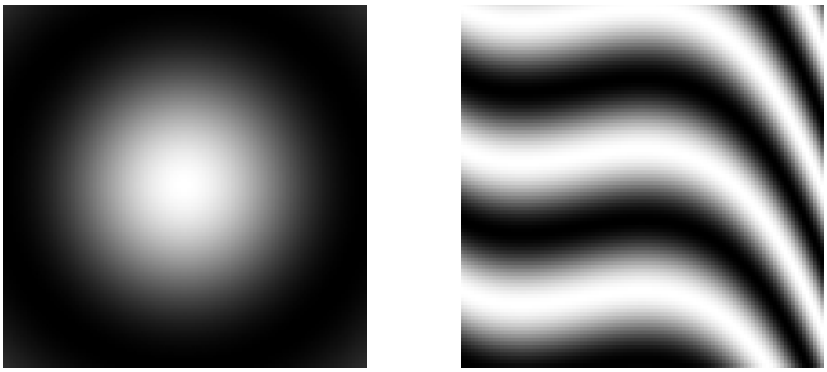
$$\begin{aligned}\text{DX-REG :} \quad \mathcal{J}(f) = \sum_{k=1}^{\mathrm{N}} \big(y_k - f(\mathbf{x_k})\big)^2 &+ \lambda_1 \|f_{\mathrm{ADD}}(\mathbf{x})\|^2_{\mathcal{H}} \\ &+ \lambda_2 \|f_{\mathrm{NL}}(\mathbf{x})\|^2_{\mathcal{H}} + \lambda_3 \|\mathcal{D}^{\mathrm{x}} f_{\mathrm{NL}}(\mathbf{x})\|^2_{\mathcal{H}},\end{aligned} \tag{4.28}$$

which has a solution closely resembling 4.23, with the addition of a regularisation term and the modification of the derivative constraint:

$$\begin{bmatrix} \mathbf{K}_{\text{ADD}}^{\top}\mathbf{K}_{\text{ADD}} + \lambda_1\mathbf{K}_{\text{ADD}} & \mathbf{K}_{\text{ADD}}^{\top}\mathbf{K}_{\text{NL}} \\ \mathbf{K}_{\text{NL}}^{\top}\mathbf{K}_{\text{ADD}} & \mathbf{K}_{\text{NL}}^{\top}\mathbf{K}_{\text{NL}} + \lambda_2\mathbf{K}_{\text{NL}} + \lambda_3\mathcal{D}^{(\text{x,x})}\mathbf{K}_{\text{NL}} \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \mathbf{K}_{\text{ADD}}^{\top} \\ \mathbf{K}_{\text{NL}}^{\top} \end{bmatrix} \mathbf{y}. \tag{4.29}$$

### 4.4.5 Simulation Example 2

The final example in this chapter aims to show how the combined F-REG and DX-REG approach formulated in the previous section can be applied to the identification of nonlinear systems in practice.



(a) $f_{\text{o},1}(y_{k-1}, y_{k-2})$ (b) $f_{\text{o},2}(u_{1,k}, u_{2,k})$

FIGURE 4.5 – The noise-free functions of $\mathcal{S}_{\text{o}}$.

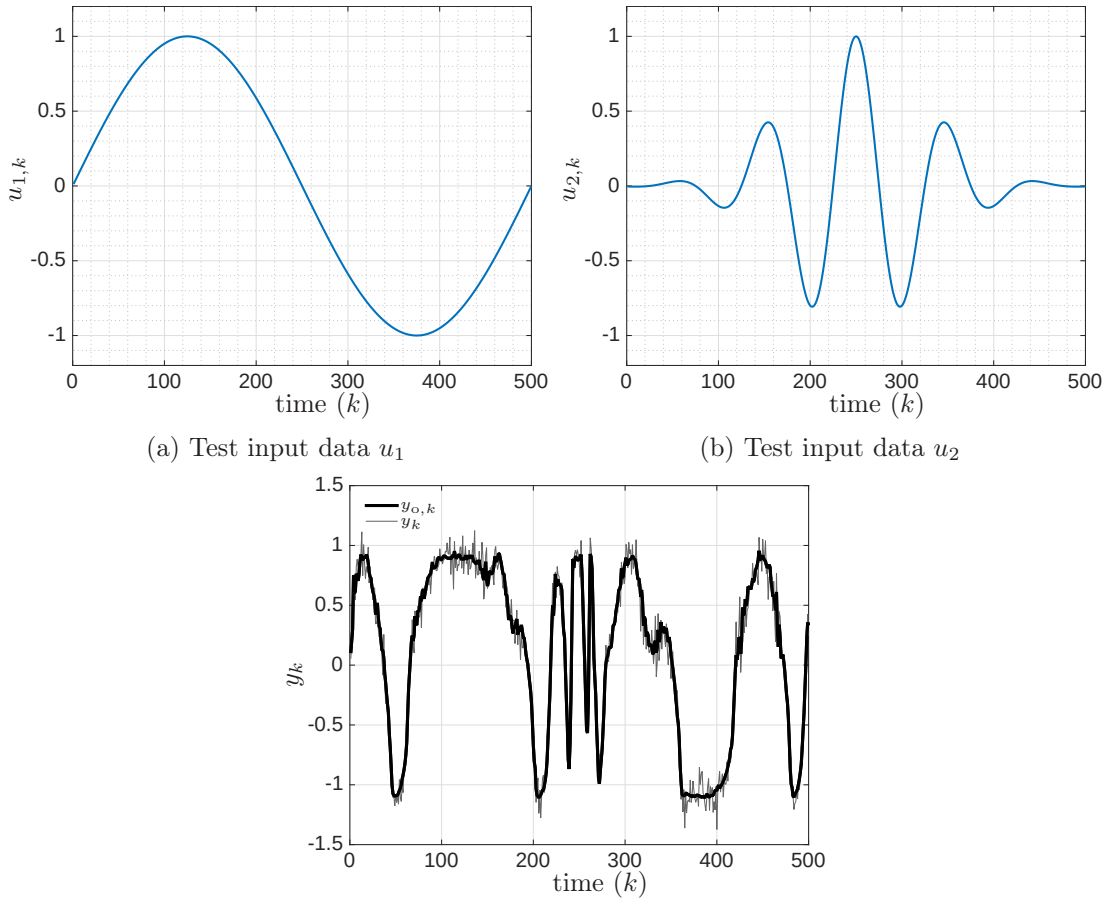
#### The Data-Generating System

The studied system of interest $\mathcal{S}_{\text{o}}$ is composed of two smooth, nonseparable functions $f_{\text{o},1}$ and $f_{\text{o},2}$ (plotted over a two-dimensional grid in Figure 4.5), and is described by the relations

$$\mathcal{S}_{\text{o}} : \begin{cases} y_k &= f_{\text{o},1}(x_{1,k}, x_{2,k}) + f_{\text{o},2}(x_{3,k}, x_{4,k}) + e_{\text{o},k}, \\ f_{\text{o},1}(x_1, x_2) &= \text{sinc}\Big( [x_1^2 + x_2^2]^{0.5} \Big) \\ f_{\text{o},2}(x_1, x_2) &= \sin\Big( 8\,[x_1 + 0.5\,x_2^2 + +0.5\,x_2^3] \Big) \\ \mathbf{x}_k &= [y_{k-1} \quad y_{k-2} \quad u_{1,k} \quad u_{2,k}]^{\top} \in \mathbb{R}^4. \end{cases} \tag{4.30}$$

To replicate the data-constraints of larger-dimensional identification problems, relatively few observations were measured given the size of the problem (N = 2000 for an input space $\mathcal{X} \subset \mathbb{R}^4$) for each experiment. The system was excited using two independently-generated, normally distributed input signals $u_{1,k}, u_{2,k} \sim \mathcal{N}(0, 0.35)$. Measurements were corrupted at the output by a white noise disturbance $e_{\text{o},k} \sim \mathcal{N}(0, 0.1)$, resulting in an SNR $\approx$ 17dB.

An estimation and validated set were generated under the same conditions, in addition to $n_{\text{mc}} =$ 100 Monte-Carlo trials. To aid visualisation of the system and the models, a deterministic test dataset was generated using a sine-wave and Gaussian pulse input, depicted in Figure 4.6. Note that this dataset was not used for validation purposes, but exclusively for visualisation.




(a) Test input data $u_1$

(b) Test input data $u_2$

(c) Test output data $y$

FIGURE 4.6 – The data-generating system $\mathcal{S}_o$.

### Experimental Procedure

Four approaches were considered for identification. In all cases it was assumed that the order of the system was *a priori* known ($n_a = 2$, $n_b = 0$, $n_u = 2$). A Gaussian RBF kernel of uniform width to construct the models for all approaches. To illustrate how the addition of a structural penalty offers an extra degree of freedom in the optimisation problem, a sequential optimisation procedure to determine the optimal model configuration was used.

1. F-REG - RKHS functional norm regularisation with the hyperparameters $(\sigma, \lambda)$ optimised using cross-validation against the noisy validation dataset.
2. DX-REG - interaction penalty with a functional norm regularisation but no bias function. Essentially, this approach is equivalent to F-REG but with an additional regularisation term. Here, values for $\sigma$ and $\lambda_1$ (corresponding to $\lambda$ in F-REG ) were taken as the optimised values of F-REG , with only $\lambda_2$ (the regularisation hyperparameter controlling the interaction penalty) optimised.
3. F-REG-ADD - RKHS functional norm regularisation with an additive separable component, equivalent to (4.29) with $\lambda_3 = 0$. Here, the kernels were constructed using the optimised $\sigma$ of F-REG , with $\log_{10} \lambda_1 = 0$ taken as an *a priori* value to ensure numerical stability and $\lambda_2$ optimised using cross-validation. In this way, the ability to use a constrained kernel

structure to reduce complexity in higher-dimensional problems can be illustrated.

4. DX-REG-ADD - and lastly, DX-REG with an additive separable bias component included - equivalent to (4.29). Here, the hyperparameter values of F-REG-ADD were used with $\lambda_3$ optimised using cross-validation. In this way, the ability of a soft structural constraint to apply a degree of tuning not achievable through the discrete kernel choices can be illustrated.

Models were trained against the estimation data of Figure 4.6, and the model hyperparameters estimated using cross-validation against the noisy validation dataset. Following determination of the optimal model configuration in each case, the models were re-estimated for the 100 Monte-Carlo trials. One-step ahead predictions against a noise-free realisation of the validation dataset were used to evaluate the performance of the models, with the FIT, MRMSE, MBIAS and MSDEV (calculated as described in (2.34)) displayed in Table 4.7. The estimated models are plotted in Figure 4.7, evaluated against the test dataset.

| | FIT | MRMSE | MBIAS | MSDEV | $\sigma$ | $\log_{10}\lambda$ |
|---|---|---|---|---|---|---|
| F-REG | 54.27% | 0.34 | 0.19 | 0.16 | 0.28 | $-5.17$ |
| DX-REG | 57.34% | 0.31 | 0.19 | 0.10 | 0.28 | $(-5.17, -1.21)$ |
| F-REG-ADD | 60.08% | 0.30 | 0.17 | 0.04 | 0.28 | $(0, -8.00)$ |
| DX-REG-ADD | 63.19% | 0.27 | 0.15 | 0.07 | 0.28 | $(0, -8.00, -0.22)$ |

TABLE 4.7 – Summarised results of the optimised models in Figure 4.7.

**Results**

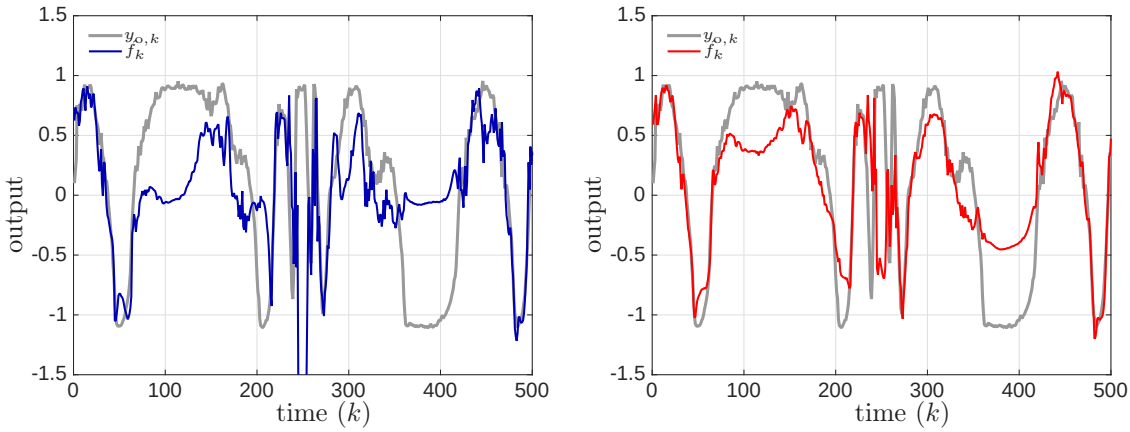
As can be seen in the results of Table 4.7, adding structural components gives a steady improvement in the performance of the estimated models, with DX-REG-ADD achieving an overall gain of $\approx 9\%$ with respect to F-REG .

Interestingly, examination of the optimised hyperparameter values illustrates just how much restrictive the functional norm penalty is - with relatively small values used (-5.17,-8.00). By constrast, the structural penalty of DX-REG affords a much less restrictive way of tuning the model properties and is therefore able to be applied more liberally (-1.21,-0.22).

In the cases of DX-REG , F-REG-ADD and DX-REG-ADD , a full optimisation over the hyperparameters would of course yield results that are at least as good as the results presented here, again illustrating just how much extra flexibility is afforded by using structural constraints in this context. In Figure 4.7, we can see that this would be desirable as, although a clear improvement is visible in the results of DX-REG-ADD in Figure 4.7b with respect to F-REG in Figure 4.7a, but there is evidently still a significant level of error in the estimate.

However, performing the optimisation in this way allows us to clearly highlight the main objective of this example: how structural penalties offer a way of acting on the model properties which is less restrictive than enforcing either smoothness or a functional norm constraint, and are therefore an appealing way of tuning complexity in nonlinear problems.

The example of this section has been deliberately chosen such that neither $f_{\mathrm{o},1}$ nor $f_{\mathrm{o},2}$ strictly correlate with the *a priori* placed by either the regularisation term or the additive component of the model. Clearly, knowledge of the exact model structure would improve the results, but




(a) F-REG estimate.

(b) DX-REG-ADD estimate.

FIGURE 4.7 – Estimated models plotted against test data.

the intention here is to show how structural constraints can be useful in cases for which perfect information is not readily available.

To state that, in any given scenario it is possible that some level of interaction between variables may be present in the system is not a strong statement. Nor is it a particularly strong statement to suggest that these interactions may not involve all variables of the model simultaeneously. What can be difficult to determine however, is how exactly these interactions occur and to what extent. Hence, the ability to continuously apply soft structural constraints on the model features could be very attractive in certain situations.

This example gives a slight insight into how structural penalities could aid the estimation of nonlinear models in practice, by offfering additional possibilities with respect to how the model complexity can be tuned. Furthermore, the formulation of D-REG allows for the incorporation of different constraints and different model components as desired - with minimal effort required from the user to formulate solutions to different problems.

## 4.5 Summary

This chapter explored the application of soft model constraints, placed through derivative regularisation terms, to different problems in nonlinear identification. It was shown that D-REG allows soft constraints to be applied in conjunction with hard constraints, which can be placed either through the kernel structure (e.g. as in section 4.3.3) or through the hyperparameter (e.g. as in section 4.3.4).

Formulating derivative penalties in an RKHS directly, as per D-REG , permits the control of a large range of model properties, which in turn opens up many possibilities with respect to how derivative regularisation-based optimisation schemes can be applied to nonlinear identification problems (as illustrated in section 4.4.3).

This chapter investigates just some of the available possibilities, namely smoothness detection in nonlinear models, structural approximation in control-oriented model structures and complexity tuning using structural penalties - such as on the separability of a model. Such penalties could be

incorporated into modelling schemes in many different ways. But as illustrated in the examples of this chapter the methods presented are most of interest when *slack* exists in the kernel definition : allowing a physical constraint to act on properties of the model otherwise difficult to access through the kernel. An example of how this could be incorporated into a practical identification scenario is illustrated in Figure

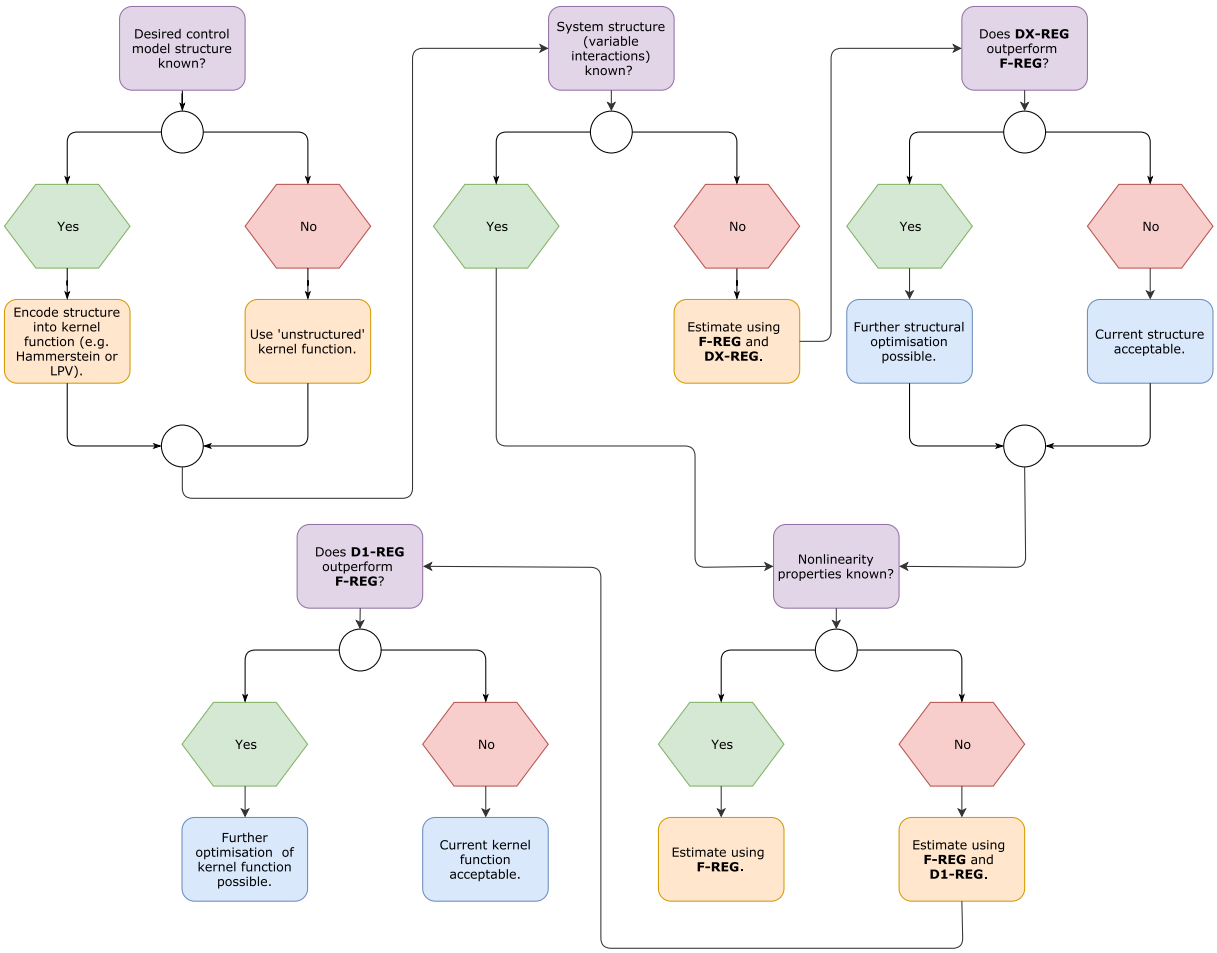



FIGURE 4.8 – Summary of the presented methods and their uses in practice.

However, many more topics have not been discussed. For example, in this chapter we have focused on the application of smooth, global constraints on the model properties. However, the approaches of SPLINES , D-REG and D-MIN can all be easily extended to allow for location application of smoothness constraints (e.g. through a weighted loss function as in [Bhujwalla et al., 2016b]). Alternatively, as D-MIN formulates penalties in $\ell_2$ rather than $\mathcal{H}$, functional derivatives can be penalised in many different ways. Whilst this by necessity involves additional effort both in the formulation and computation of a model, such methods open many possibilities to the user (e.g. variable selection [Rosasco et al., 2010], order selection [Duijkers et al., 2014], changepoint detection [Lauer et al., 2012]).

Arguably, the full scope of such methods is not yet fully understood. This may in part be to the relative novelty of such approaches in nonlinear identification, or due to the relative difficulty in incorporating the SPLINES method into different problems of interest. Nonetheless, in addition

to all the topic mentioned previously, certain areas of potential interest have been discussed in the literature.

For example, [Ramsay and Silverman, 1997] includes a discussion on the use of derivatvie constraints based on dynamical systems, i.e. regularisation terms that enforce a model that approximates the system in the way most adhering to a particular linear differential equation. This is a potentially very exciting path, which could offer engineers a way of directly incorporating physically meaningful constraints into the optimisation problem - and perhaps help to bridge the gap between control theory and nonlinear modelling.

Unfortunately, such topics are beyond the scope of this thesis. However, in the following chapter, the D-REG method will be applied to an industrial problem: modelling internet traffic for a European internet service provider. It will be shown that both a kernel-based identification approach and a smoothness-enforcing regulariser offer the user a straightforward framework for the development of accurate models in real-world problems.

# 5

# Application to Real Data

*In the final chapter of this thesis, the methods presented will be applied to an industrial problem, namely that of modelling 'Autonomous System'-level internet traffic.*

## 5.1 Introduction

Over the course of this thesis, it has been aimed to illustrate how kernel methods (whether they be LS-SVMs, GPs, RKHS and Sobolev space methods or otherwise) provide an appealing framework for nonlinear identification problems.

Amongst their many characteristics, perhaps the most interesting of their strengths from an identification perspective is their versatility. This versatility is apparent not only in that way that they can be applied to different structural problems (e.g. LPV and block-oriented models) but also in how a very similar problem setup (model class, optimisation criterion, model definition and solution) can be applied to many different types of nonlinear problems, with relatively few modifications required.

Similarly, we have tried to illustrate how incorporating derivative regularisation terms into the cost-function can further extend the scope of such methods, by changing the way in which the model properties can be tuned, in certain cases offering additional accuracy and physical insight from the estimated models.

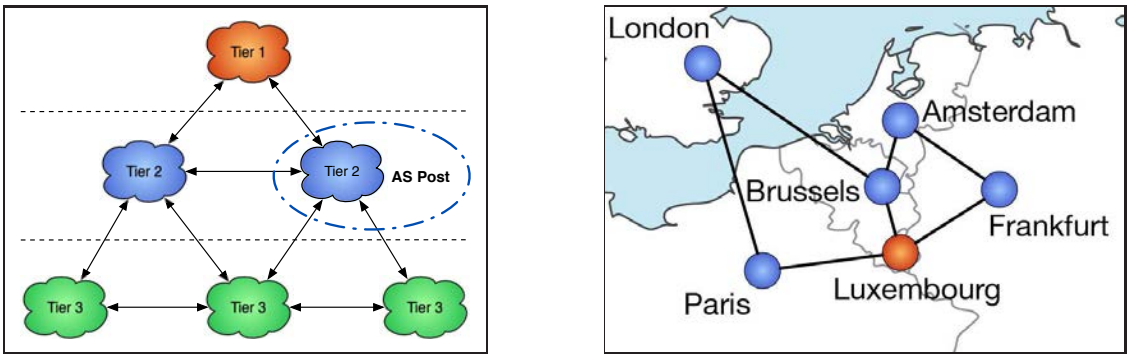
In this chapter, a real-data example will be provided to further illustrate the above points. Using data obtained from an industrial partner, a kernel-based nonlinear model will be developed for the problem of modelling internet traffic. This chapter documents the first part of an ongoing collaboration with Post Luxembourg, a European internet service provider. The aim of the study has been to collect and interpret traffic data, for subsequent usage in network maintenance, analysis and optimisation. This has involved several stages including:

- Data acquisition and analysis [Grandemange et al., 2017c],
- Time-series modelling [Grandemange et al., 2017b],
- And kernel-based modelling [Bhujwalla et al., 2017a, Grandemange et al., 2017a].

The methods presented here are currently in the final stages of testing and development, with online deployment envisaged within the coming months. It is hoped that, with such a resource

now available, the work performed up to this point will provide a basis for future developments in the application of nonlinear identification methods to internet traffic modelling.

## 5.2 The Post Luxembourg Network




(a) *Post* in the three-tier internet hierarchy (b) *Post's* PE routers

FIGURE 5.1 – The Post Luxembourg network.

The data used in this study was obtained with the help of *Post Luxembourg*, in collaboration with the *Centre de Recherche en Automatique de Nancy* (CRAN). *Post* is a European *internet service provider* ISP which, at the end of 2014, provided fibre-optic to 30,000 and copper connections to 40,000 customers in Luxembourg. Because of the transit it gives to other ISP, it can be classified as a Tier-2 operator within the Three-Tier internet hiercharchy (Figure 5.1a).

*Post* is particular in that, unlike many other ISPs, there is significant traffic both coming into and going out of the network. This is because *Post* also provides hosting solutions to its customers - meaning that it has the particularity of acting both as an ISP and a *content delivery network* (CDN).

The centrality of *Post* within the *European Blue Banana* [Brunet, 1989], and role of Luxembourg as a financial centre within the global market, result in significant traffic flow through the observed network. As such, it has *physical edge* (PE) routers in different *internet exchange points* IXPs - as illustrated in Figure 5.1b.

### 5.2.1 AS-Level Analysis

The internet can be thought of as a vast collection of interconnected sub-network, spread across the globe, passing information at high speed. Although it can be compared to other types of networked systems, such as electrical power or water distribution, its rapid growth and constantly evolving structure means that managing internet traffic presents a unique set of challenges.

To date, most efforts to understand internet traffic have focused on either link-level or flow-level analysis, e.g. [Barakat et al., 2002, Cortez et al., 2012]. This is, to a certain extent, informative as it allows ISPs to meet the physical demands placed on the network. However, this is also a natural limitation imposed by a lack of availability of data at a larger scale - which, in turn,

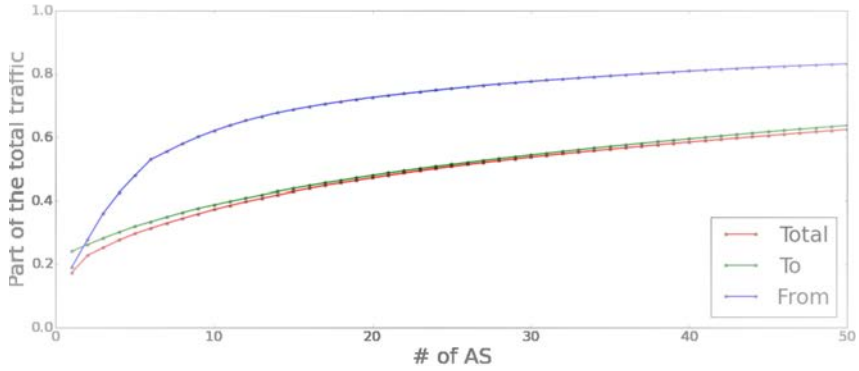



FIGURE 5.2 – Cumulative traffic per AS on the *Post Luxembourg* network.

impedes our ability to understand how individual components interact with each other and how this relates to the behaviour of the network as a whole.

Recently however, attention has turned to other types of analyses that better reflect the needs of ISPs. For example, rather than looking primarily at the net traffic through individual channels, we can categorise traffic over many channels by its source and destination.

In this scenario, our interest is to analyse the traffic coming from and going to different *autonomous systems* (AS). An AS is a group of one or more IP addresses known to correspond to a particular agent on the network, which could be, for example, a website such as Google or Facebook. Whilst such an approach may not have been of interest in the past - where traffic flow was accounted for by many different AS - as usage patterns change the incentive to move towards an AS-level analysis grows stronger.

Whilst there are approximately 54,000 AS currently active on the network [Bates et al., 2016], with an additional 240 new AS created each month [Yang and Rong, 2015], traffic is overwhelmingly dominated by a very small group of AS. For example, analysis of the *Post Luxembourg* network in [Grandemange et al., 2017c] showed that 60% of the total traffic coming into the Post network is from just 10 AS, as illustrated in Figure 5.2. Hence, this small group of AS (including well-known sites such as Amazon, Facebook, Google and Netflix) is likely to dictate network traffic - and optimisation of the network performance would be greatly facilitated by an understanding of how these AS behave.

Another consequence of this is that, as traffic is skewed towards a small group of AS, unexpected behaviour on any one AS could significantly affect the network as a whole. As an example, consider Apple Inc. (AS 6185). In September 2016, Apple released software updates for their phones, sparking a unexpected surge of traffic from their AS. This resulted in a saturation of links across the network, detrimentally affecting all other network activity (Figure 5.3). Although such extreme situations are rare, this case does show how a single AS can have a massive impact on network performance.

Moreover, anomalous behaviour is not only caused by unforeseen usage. Malicious attacks on an AS are common. And clearly, accurate forecasting of behaviour at an AS-level could help to quickly detect attacks - preventing unnecessary load on the network, and in some cases, the loss of sensitive data.

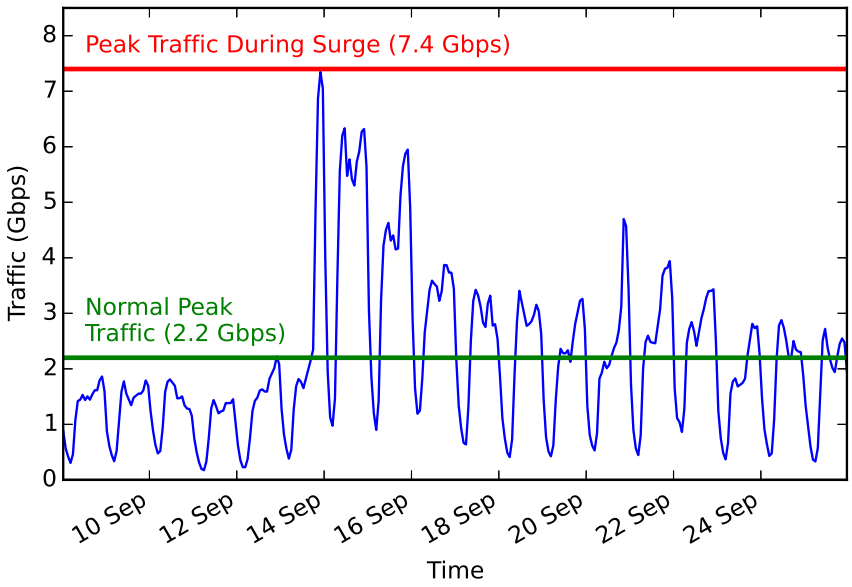



FIGURE 5.3 – Apple (AS 6185) traffic during iOS10 release (08-25/09/2016)

### 5.2.2 The Need for Models

Clearly then, an understanding of how traffic flows between a network and individual AS could be very useful to ISPs both in maintaining and optimising the network. However, whilst relatively few AS are significant - with respect to the total number of active AS on the network - there are still many important AS. Furthermore, the number of significant AS, which AS are significant and how these AS behave all change over time, making analysis very challenging.

In this context, data-driven modelling could be vey helpful - both in aiding understanding and even removing the need to understand network usage in certain cases. Accurate traffic flow models could, for example:

- Aid analysis of individual AS and the network as a whole.
- Provide an alternative to the costly storage of vast quantities of largely uninformative historical data.
- Help anticipate future load on the network.
- And aid detection of anomalous behaviour.

But, modelling traffic flow is not straightforward - with constraints placed on the modelling process both by the characteristics of the systems, and the requirements of the user:

- Whilst historical data should be informative about future observations of the system, it is not clear what other factors may be important.
- And, systems are known to change over time - both in their physical structure, and due to short-term and long-term shifts in usage patterns [Grandemange et al., 2017c].
- As many systems need to be modelled, it is impractical for network engineers to spend significant time developing models for any one particular system: it should be possible to use a flexible, generic framework which can be readily adapted to different systems as needed, with minimal interference.
- As models could be used for many different purposes, it should be possible either to develop a 'one-size-fits-all' method, or to easily adapt the modelling process to meet different requirements as necessary.

In the following section, two different approaches to the traffic modelling problem will be proposed, with a comparison of the two methods presented in section 5.4.

### 5.2.3 Data Acquisition

As mentioned in the previous section, one obstacle to AS-level analysis is access to data. However, in [Jiang et al., 2010, Grandemange et al., 2017c], it has been shown how such data can be acquired.

The data used for this study was obtained with the aid of Netflow V9 [Claise, 2004]. Netflow is a protocol developed by [Claise, 2004], and is widely used for *Distributed Denial of Service* (DDoS) attack prevention. By observing flow through the seven different operational PE routers connected to *Post* (some of which depicted in Figure 5.1b), flow into the network can be measured.

Due to the sheer quantity of flow data, measuring every packet passing through the network is infeasible, and hence, only one packet per thousand is recorded. Following aggregation of the data, the resulting measurements are rescaled, to give a more accurate indicator of flow levels through the network. The impact of this sampling rate on the resulting observations is discussed in greater detail in [Grandemange et al., 2017c]. An example of the obtained data can be seen in Figure 5.4.

### 5.2.4 Premiminary Analysis

In this study, measurements from three AS over a period of 80 days are considered. The chosen AS (LUCIX, Hinet and Netflix) are known to represent a significant proportion of the overall network traffic. Furthermore, is known that all three exhibit different behavioural patterns, and as such should provide a reliable test of the modelling approaches presented in subsequent sections. As both traffic coming into and going out of the *Post* network are important (as discussed in Section 5.2), in each case the direction of greatest flow is taken.

If each AS is considered as an unknown system $\mathcal{S}_o$, the obtained measurements can be considered as N observations of each system at different time-instances :

$$\mathcal{Z}_{\mathrm{N}} = \{(t_1, y_1), \ldots, (t_k, y_k), \ldots, (t_{\mathrm{N}}, y_{\mathrm{N}})\}\,, \tag{5.1}$$

with $t_k \in \mathbb{R}_{\geq 0}$ the time-stamp of each output traffic measurement $y_k \in \mathbb{R}\ \forall k = 1, \ldots, \mathrm{N}$. As the dependencies of the system are unknown, no input variable is measured.

Although the data observed in Figure 5.4 is clearly nonsmooth, it is reasonable to assume that between samples (taken at intervals of $T_s = 300s$) the underlying true system in each case is largely smooth under normal operating conditions.

Furthermore, despite the dynamics of the system being unknown, there is clear periodicity in the output measurements, as illustrated in Figure 5.4. Hence, a modelling paradigm capable of accounting for the time-periodic nature of each system will likely be a much better predictor of future load.

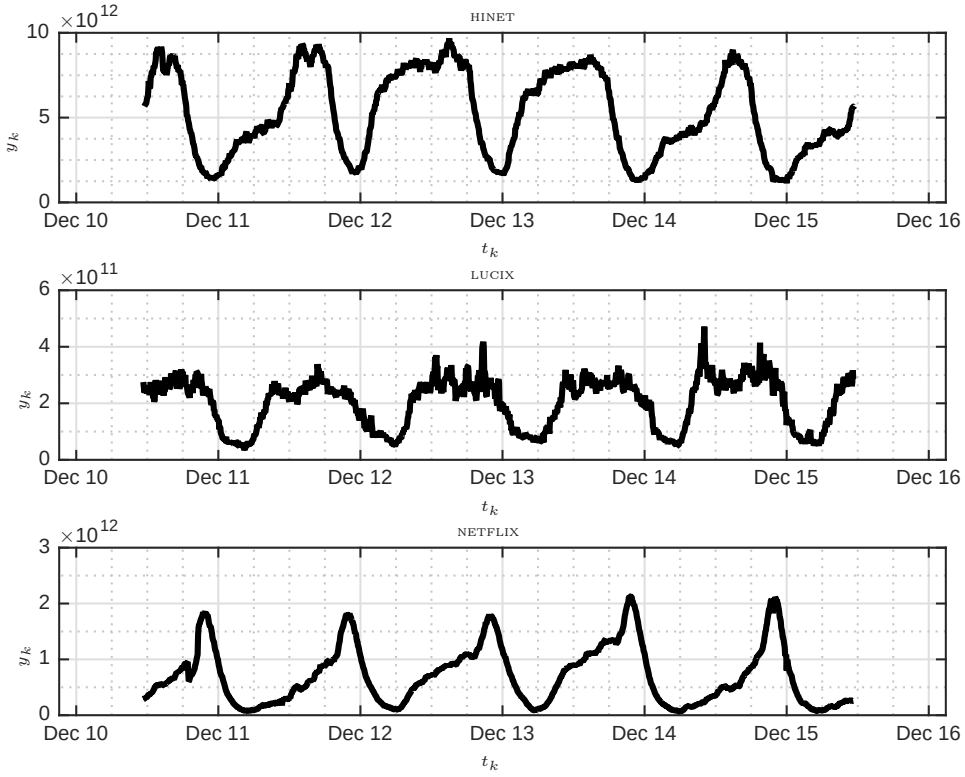



FIGURE 5.4 – A five day sample of the observed flow data.

## 5.3 Traffic Modelling

Based on the discussion of the previous sections, two strategies for the AS-level traffic modelling problem will now be presented.

1. The first is an existing method from the literature, which although unstudied in the specific context of traffic modelling, has been succesfully applied to many similar problems and can be considered a reasonable *benchmark* method with which the performance of other methods can be assessed.

2. The second is a *system identification*-inspired kernel-based approach, developed using the results presented in the previous chapters.

### 5.3.1 Time-Series Analysis

In this section, we will consider the modelling problem from a time-series perspective. The field of *Time-Series Analysis* is devoted to the estimation of numerical models for input-free data, with particular consideration of time-varying systems [Young, 2011, Hasselmann et al., 1963]. Consequently, time-series modelling techniques are well-suited to the network-traffic modelling problem [Cortez et al., 2012].

### Dynamic Harmonic Regression

Although no input to the system is measured, and the dependencies of the system are unknown, many AS exhibit clear periodic tendencies - as discussed in section 5.2.4. Accordingly, we propose here the use of the *Dynamic Harmonic Regression* (DHR) approach [Young et al., 1999] :

$$\mathcal{M}_{\text{DHR}}: \quad y_k = T_k + C_k + S_k + e_k, \quad e_k \sim \mathcal{N}(0, \sigma^2) \tag{5.2}$$

The different components of $y_k$ above all characterise some aspect of the time-periodic behaviour. $T_k$ is a low-frequency trend component, that can also allow for non-stationarity in the signal $y_k$. $C_k$ and $S_k$ model *cyclical* and *seasonal* variations in $\mathcal{S}_0$ and can be defined as :

$$\begin{aligned} C_k &= \sum_{i=1}^{R_c} \{a_{i,k} cos(\omega_i k) + b_{i,k} sin(\omega_i k)\} \\ S_k &= \sum_{i=1}^{R_s} \{\alpha_{i,k} cos(f_i k) + \beta_{i,k} sin(f_i k)\}. \end{aligned} \tag{5.3}$$

The time-dependency of $\mathcal{M}_{\text{DHR}}$ is characterised by a stochastic evolution of the model parameters $a_{i,k}$, $b_{i,k}$, $\alpha_{i,k}$, $\beta_{i,k}$, described by a *Generalised Random Walk* (GRW) process [Jakeman and Young, 1984] :

$$\begin{gathered} \mathbf{x}_{i,k} = \mathbf{F}_i \mathbf{x}_{i,k-1} + \mathbf{G}_i \eta_{i,k-1}, \; i = 1, \ldots, R_c + R_s + 1 \\ \mathbf{F}_i = \begin{bmatrix} \alpha & \beta \\ 0 & \gamma \end{bmatrix} \quad \mathbf{G}_i = \begin{bmatrix} \delta & 0 \\ 0 & 1 \end{bmatrix}. \end{gathered} \tag{5.4}$$

The state vector $\mathbf{x}_{i,k} = [l_{i,k} \; d_{i,k}]^\top$ defines the changing level ($l_{i,k}$) and slope ($d_{i,k}$) of the parameter. The variables $\alpha$, $\beta$, $\gamma$, $\delta$ determine the evolution of $\mathbf{x}_{i,k}$ and noise model is denoted by $\eta_{i,k-1}$.

The approach can essentially be summarised as a spectral Fourier decomposition of the observed data, where the Fourier coefficients (representing the model parameters) are allowed to vary in time according to changes in the system. Although DHR is not widely-used in traffic modelling problems, many examples can be found of its application to time-varying periodic systems, for example [Gonzàlez and Moral, 1997, Tych et al., 2002].

### DHR: Implementation Notes

To estimate a model using DHR, certain choices in its configuration must be made. The theory behind these choices, and a comprehensive description of how to make these choices, can be found in [Young et al., 1999, Young, 2011]. However, for the benefit of the reader, this process is briefly outlined here, in the context of the AS-level analysis problem discussed in this paper. The models in question were developed with the help of the **CAPTAIN Toolbox** for MatLab [Young and Taylor, 2012].

- **Pre-processing :** To improve the accuracy of the models, it is advisable to standardise the measurements prior to estimation, by removing the mean and normalising the variance of the data. Additionally, downsampling can aid longer-term forecasting, by reducing the influence of higher-frequency terms in the model.

- **Spectral Bases :** To estimate the frequency bases of the system ($\omega_i$, $f_i$ in (5.3)), the `arspec` routine from the CAPTAIN Toolbox can be used. In practice, it is more effective to estimate these bases with respect to a high-order AR model, fitted to the observed data.
- **Hyperparameters :** The model further depends on the configuration of certain hyperparameters :
  1. **Time-Varying Parameters (TVPs) :** To determine the evolution in time of the model parameters, the parameters of $\mathbf{F}_i$ (5.4) must be defined - and can be chosen using suitable prior knowledge of the system, or as in this case, by using cross-validation to determine the optimal configuration.
  2. **Noise-Model Hyperparameters :** The noise-model hyperparameters (in $\mathbf{G}_i$ (5.4)) can be optimised using the `dhropt` routine, available in the CAPTAIN Toolbox.
- **Estimation :** Finally, given a suitable configuration, $\mathcal{M}_{\text{DHR}}$ can be estimated from the data using the `dhr` routine, from the CAPTAIN Toolbox.

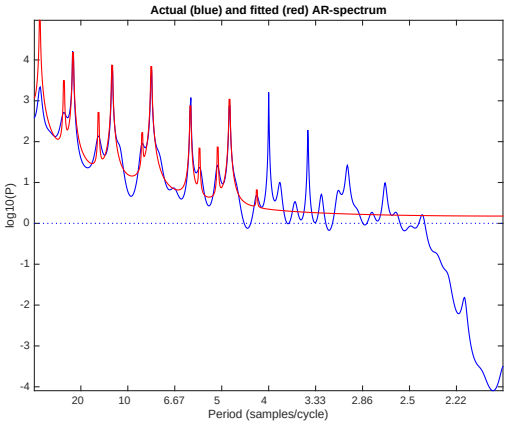



FIGURE 5.5 – Estimated spectrum (blue) and model spectrum (red) for the Netflix network

### 5.3.2 A Kernel-Based Approach

System identification as a subject in general prides itself not only on its ability to develop good models using sophisticated techniques, but also on taking a thorough and measured approach to the entire modelling process - as can be seen by the comprehensive nature of fundamental texts in the literature such as [Ljung, 1999, Pintelon and Schoukens, 2012] and many others.

In this section we would like to show that kernel-based identification is no different, and can draw from the identification literature in the same manner as any other modelling framework. Consequently, we will present a step-wise treatment of the modelling process, with all the necessary results from the previous chapters included for the convenience of the reader.

#### Data

In identification, the first important question is almost always regarding the utility of the data available. In this case, the measurements obtained are time-series measurements, which could be

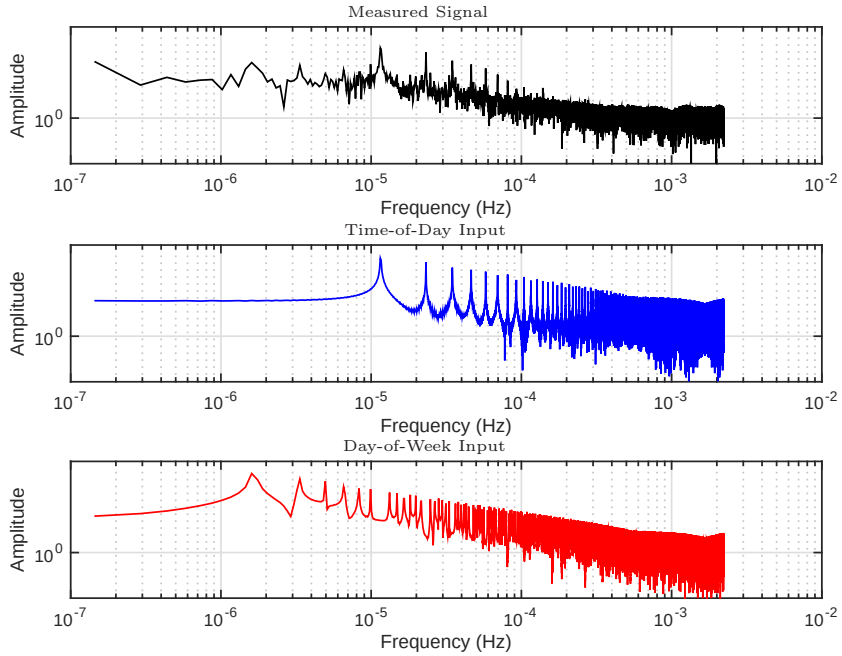



FIGURE 5.6 – Netflix signal spectrum against *time-of-day* and *day-of-week pseudo-input* signals

used, but as we know that the system exhibits strong periodic tendencies this can be incorporated into the model in a similar fashion to the DHR method of the previous section.

Visual inspection of the data suggests strong daily and weekly periodic trends, which is confirmed by examination of the spectra of the measurements. In Figure 5.6, a clear correlation between the spectrum of the Netflix data and a *time-of-day* and *day-of-the-week* signal is visible. Hence, we can reformulate the dataset of (5.1) as

$$\tilde{\mathcal{Z}}_N = \{(\mathbf{u_1}, y_1), \ldots, (\mathbf{u_k}, y_k), \ldots, (\mathbf{u}_N, y_N)\}, \tag{5.5}$$

where :

$$\begin{aligned} \mathbf{u_k} &= [\, u_1(t_k) \quad u_2(t_k) \,]^\top \\ u_1(t_k) &= \textit{time of day} \text{ at } t_k, \\ u_2(t_k) &= \textit{day of the week} \text{ at } t_k. \end{aligned} \tag{5.6}$$

As more rigorous approach to their definition could be taken if desired (such as using the `arspec` routine discussed in section 5.3.1), but it will be shown later that this is largely sufficient.

### Model Structure

As discussed in Chapter 1, another important step in the identification process is the definition of a model structure and model class. In this case, the dependency on the periodic components is *a priori* unknown so we would like to avoid constraining this component of the model as much as possible. By contrast, whilst the system clearly exhibits some sort of dynamical behaviour, it is desirable to keep the modelling of the dynamics as simple as possible to reduce the risk

of overfitting and facilitate reproduction of the model onto other systems. Therefore, a simple first-order linear dynamical term is used.

$$\begin{aligned} \mathcal{M}_{\text{SID}}: \qquad f(\mathbf{x}_k) &= f_{\text{LIN}}(\mathbf{x}_k) + f_{\text{NL}}(\mathbf{x}_k) \\ &= \theta\, x_{1,k} + f_{\text{NL}}(x_{2,k}, x_{3,k}) \\ \mathbf{x}_k &= \begin{bmatrix} y_{k-1} & u_{1,k} & u_{2,k} \end{bmatrix}^\top. \end{aligned} \tag{5.7}$$

Of course, the complexity of the dynamical term could be increased, but considering such a simple case will also have the advantage of reducing the complexity of the model selection problem.

To define a model class corresponding to the nonlinear component of the model, a nonparametric kernel representation is used:

$$f_{\text{NL}}(\mathbf{u_k}) = \sum_{i=1} \alpha_i k_{\mathbf{u_i}}(\mathbf{u_k}). \tag{5.8}$$

As the input signals span are regularly spaced periodic signals, they define a discrete, finite set of values. Therefore, without loss of generalisation, the nonparametric representation of (5.8) can be reduced to a finite representation:

$$f_{\text{NL}}(\mathbf{u_k}) = \sum_{i=1}^{\text{M}} \alpha_i k_{\mathbf{u_i^*}}(\mathbf{u_k}), \tag{5.9}$$

where $\mathbf{u}^* \in \mathbb{R}^{\text{M}\times 2}$ is a two-dimensional grid of values spanning the input space, with

$$\begin{aligned} \mathbf{u_1^*} &= \begin{bmatrix} 0 & 5 & 10 & \cdots & 1435 & 0 & \cdots & 1435 \end{bmatrix}^\top \\ \mathbf{u_2^*} &= \begin{bmatrix} 0 & \cdots & 0 & 1 & \cdots & 1 & 2 & \cdots & 6 \end{bmatrix}^\top \end{aligned} \tag{5.10}$$

and M = 2016.

The model class of $f_{\text{NL}}$ depends on the choice of kernel function K. As in previous chapters, we will consider the Gaussian RBF kernel:

$$\begin{aligned} \text{K}_i(u_{i,j}, u_{i,k}) &= \exp\left\{-\frac{\| u_{i,k} - u_{i,j} \|_2^2}{\sigma_i^2}\right\} \\ \text{K}(\mathbf{u_j}, \mathbf{u_k}) &= \prod_{i=1}^{n_u} \text{K}_i(u_{i,j}, u_{i,k}), \end{aligned} \tag{5.11}$$

which defines a model class of smooth nonlinear functions subject to selection of the kernel hyperparameters $\sigma = \begin{bmatrix} \sigma_1 & \sigma_2 \end{bmatrix}^\top$.

### Optimisation Criterion

Another important part of the identification process is the selection of a suitable optimisation criterion, through which the model parameters ($\theta \in \mathbb{R}$ and $\alpha \in \mathbb{R}^{\text{M}}$) can be estimated. Many different optimisation criteria have been discussed in this thesis, all with different strengths and weaknesses. Here we will take just one example - the gradient minimisation algorithm D1-REG :

$$\mathcal{J}_{\text{D1-REG}}(f) = \|\mathbf{y} - f(\mathbf{x})\|_2^2 + \lambda \sum_{i=1}^{n_u} \| \partial_{u_i} f_{\text{NL}} \|_{\mathcal{H}}^2. \tag{5.12}$$

To see why, we refer again to the data in Figure 5.4. Clearly, the data is smooth over much of the input space - in the sense that inference can reasonably be made about traffic loads from one instance to another nearby. However, we can also see examples of sharp shifts in flow levels for each system. These shifts could be due to noise in the system or anomalous behaviour. But they could also be part of the normal operating conditions of the system, caused perhaps by simple events such as the beginning and end of the working day, lunchtimes or any other event. As illustrated in section 4.2, D1-REG is capable of placing smoothness constraints in a soft manner that preserves the ability of a model to capture sharp changes when such changes appear informative. Therefore, such an optimisation criterion may be able to help distinguish between disturbances and the true behaviour of a system.

As presented in section 3.4, the model parameters can be estimated according to

$$\begin{bmatrix} \mathbf{X}^\top\mathbf{X} & \mathbf{X}^\top\mathbf{K} \\ \mathbf{K}^\top\mathbf{X} & \mathbf{K}^\top\mathbf{K} + \lambda\mathcal{D}\mathbf{K} \end{bmatrix} \begin{bmatrix} \theta \\ \alpha \end{bmatrix} = \begin{bmatrix} \mathbf{X}^\top \\ \mathbf{K}^\top \end{bmatrix} \mathbf{y}. \tag{5.13}$$

To preserve the simplicity of the solution, we neglect consideration of a bias function here, yielding the following expressions for the matrices $\mathbf{K} \in \mathbb{R}^{N\times M}$, $\mathcal{D}\mathbf{K} \in \mathbb{R}^{M\times M}$ and $\mathbf{X} \in \mathbb{R}^{N}$ :

$$\begin{aligned} \{\mathbf{K}\}_{j,k} &= \mathrm{K}(\mathbf{u_j}, \mathbf{u_k}) \\ \{\mathcal{D}\mathbf{K}\}_{j,k} &= \sum_{i=1}^{n_u} \frac{\partial^2 \mathrm{K}(\mathbf{u_j}, \mathbf{u_k})}{\partial u_{i,j} \partial u_{i,k}} \\ \{\mathbf{X}\}_{j,k} &= \begin{cases} y_{j-k} & j > k \\ 0 & j \leq k \end{cases} \end{aligned} \tag{5.14}$$

**Model Selection**

The solution of (5.13) is defined for a particular kernel function and regularisation hyperparameter $\lambda$. Use of D1-REG allows us to remove the kernel hyperparameter from the optimisation problem, and instead place the smoothness constraint on the regularisation term. Using the results of section 3.4.3, $\sigma$ can be selected with respect to the maximum spacing of the observations such that

$$\begin{aligned} \sigma &= \sqrt{n_u}\rho_k \left[\, \Delta_{u_1} \quad \Delta_{u_2} \,\right]^\top \\ \left[\, \Delta_{u_1} \quad \Delta_{u_2} \,\right]^\top &= \left[\, 5\mathrm{mins} \quad 1\mathrm{day} \,\right]^\top. \end{aligned} \tag{5.15}$$

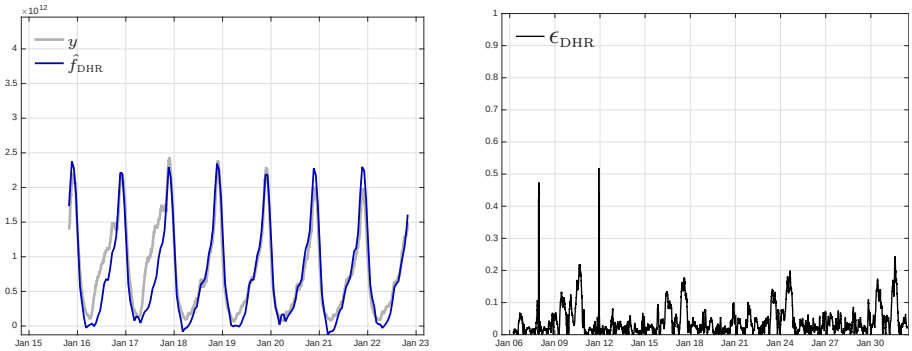
The kernel density parameter $\rho_k$ is chosen as $\rho_k = 2$ - following the guidelines proposed in section 3.4.3, yielding

$$\begin{aligned} \sigma &= 2\sqrt{2}\left[\, 5 \quad 1 \,\right]^\top \\ &\approx \left[\, 14.2 \quad 2.8 \,\right]^\top. \end{aligned} \tag{5.16}$$

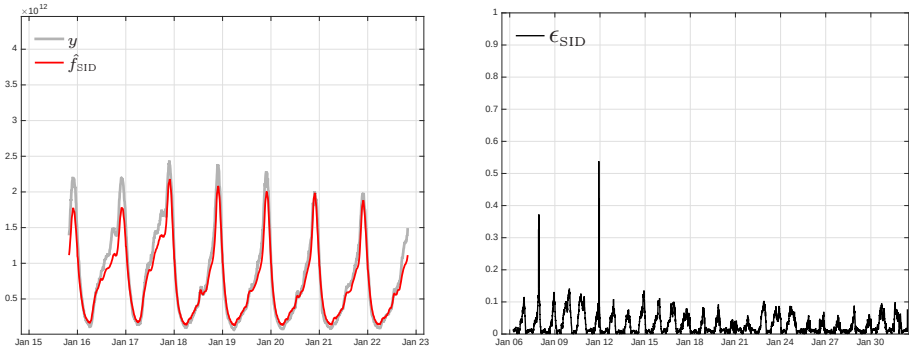
The regularisation hyperparameter is then the only part of the model that needs to be tuned, and can be chosen using one of the many techniques available in the literature. Here, we will use cross-validation against obtained measurements to ensure a reasonable value of $\lambda$ is obtained.

## 5.4 Model Evaluation

To determine the suitability of the modelling strategies proposed in section 5.3, their performance against traffic data discussed in section 5.2.4 is evaluated in this section.




(a) $\mathcal{M}_{\text{DHR}}$: 5 day sample. (b) $\mathcal{M}_{\text{DHR}}$: relative error.




(c) $\mathcal{M}_{\text{SID}}$: 5 day sample. (d) $\mathcal{M}_{\text{SID}}$: relative error.

FIGURE 5.7 – NETFLIX results.

### 5.4.1 Experimental Procedure

Each dataset contains N = 22,880 observations, spanning the period November 2015 to January 2016.

In each case, the datasets are split up into three subsets of equal length (an estimation set, a validation set and test set), with the models estimated using a two-step procedure :

- The estimation dataset is used to estimate the parameters, with the validation dataset used to determine the optimal hyperparameters and model configuration for each approach.
- Following the determination of suitable hyperparameters, models are re-estimated over the *seen* estimation and validation datasets, and then evaluated against the *unseen* test dataset.

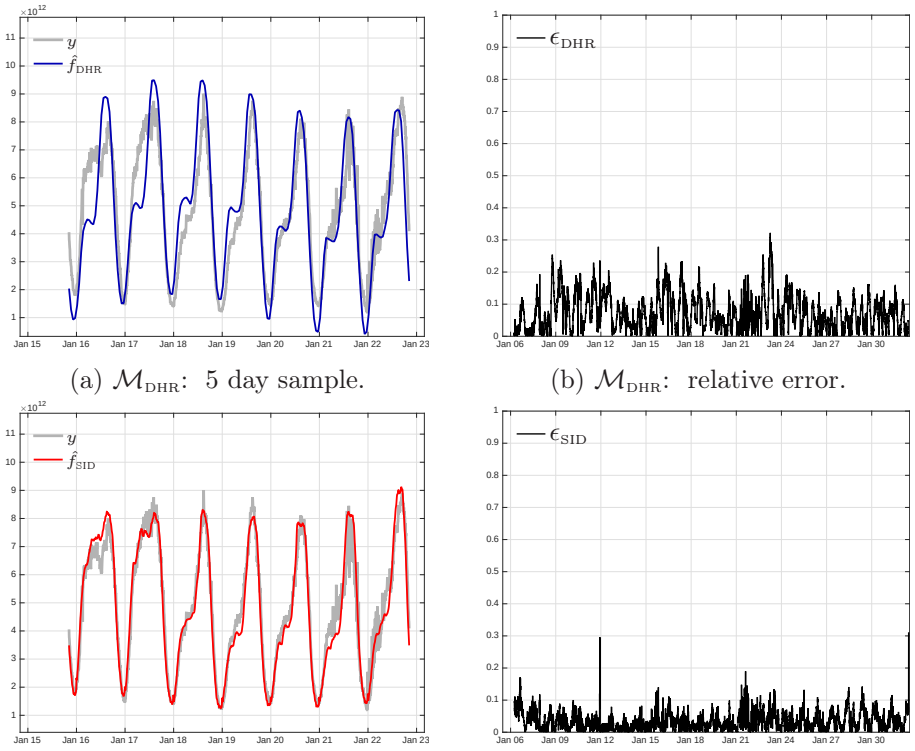
Two models are estimated for each dataset :

1. $\mathcal{M}_{\text{DHR}}$ : estimated according to the procedure described in section 5.3.1, with the choice of Fourier bases and the evolution of the TVPs optimised against the validation dataset.
2. $\mathcal{M}_{\text{SID}}$ : estimated according to the procedure described in section 5.3.2, with the regularisation hyperparameter optimised against the validation dataset.

Values for the fit of each model against the unseen test dataset are given in Table 5.1, estimated as

$$\text{FIT} = 100 \cdot \left(1 - \frac{\|y - \hat{y}\|^2}{\|y - \bar{y}\|^2}\right)\%. \tag{5.17}$$

To illustrate the performance of the models, a five day sample of the models estimated for each




(a) $\mathcal{M}_{\text{DHR}}$: 5 day sample.

(b) $\mathcal{M}_{\text{DHR}}$: relative error.

(c) $\mathcal{M}_{\text{SID}}$: 5 day sample.

(d) $\mathcal{M}_{\text{SID}}$: relative error.

FIGURE 5.8 – HINET results.

dataset are plotted in Figures 5.7-5.9, with the model errors displayed for the whole dataset using a *relative error* metric to facilitate comparison:

$$\epsilon_k = \frac{\|y_k - \hat{y}_k\|}{\max\{y\}}. \tag{5.18}$$

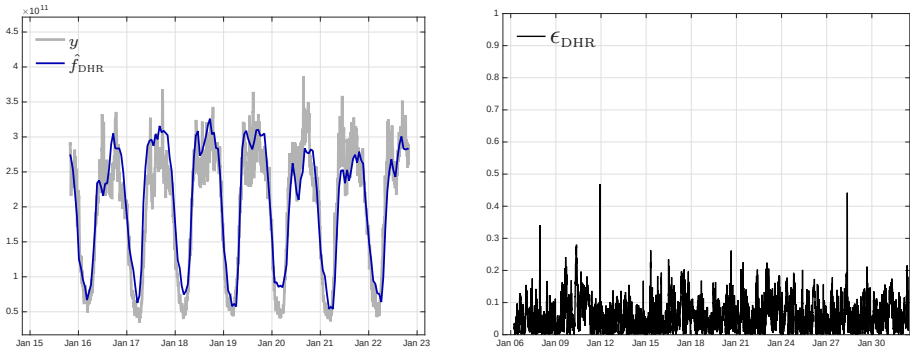
### 5.4.2 Results

The results of Table 5.1 indicate that the performance of both techniques is largely satisfactory. This is further emphasised in Figures 5.7-5.9, where clearly the periodic behaviour of the true data is well-captured in both cases.

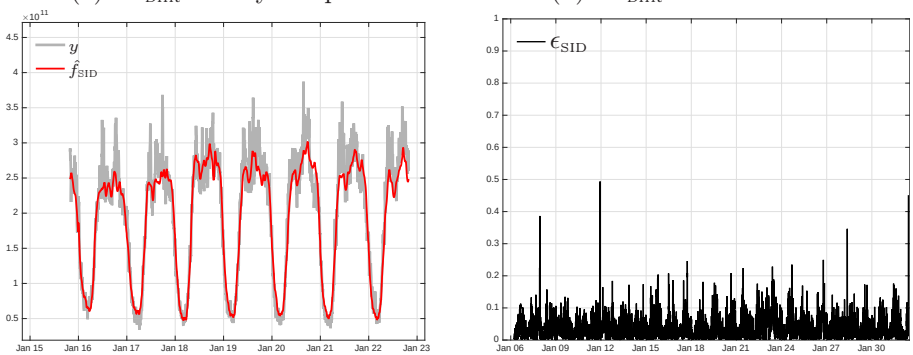
In Figure 5.7, the error bars for each method show clearly the presence of certain outliers in the models around early January. In fact these are due to three separate incidents, each resulting in a loss of measurements at this time period. Here the potential utility of the models in detecting unanticipated behaviour on the network is clearly demonstrated.

As the fit scores of Table 5.1 show, over a shorter window (such as the three-month window used in this example) $\mathcal{M}_{\text{SID}}$ typically outperforms $\mathcal{M}_{\text{DHR}}$, as it benefits from the increased flexibility of a nonparametric approach - and the freedom in the choice of kernel function.

However, $\mathcal{M}_{\text{DHR}}$ is arguably more suited to prediction over a longer period (though not necessarily to longer-term forecasting), as it is capable of adapting to changes in the system over time. This




(a) $\mathcal{M}_{\text{DHR}}$: 5 day sample. (b) $\mathcal{M}_{\text{DHR}}$: relative error.




(c) $\mathcal{M}_{\text{SID}}$: 5 day sample. (d) $\mathcal{M}_{\text{SID}}$: relative error.

FIGURE 5.9 – LUCIX results.

can be seen in how $\mathcal{M}_{\text{SID}}$ struggles to model the higher loads observed during Janury, whereas as these are well-anticipated in the DHR approach.

In the case of $\mathcal{M}_{\text{SID}}$, adapting to time-variations is less straightforward, and would likely require the offline re-estimation of the model. In practice this is likely to be feasible, as significant changes in behaviour are relatively infrequent (with respect to computational load). It is possible that information regarding the time-varying nature of the system could be incorporated into the modelling framework, for example through the choice of a more suitable kernel function [Pillonetto, 2008].

Nonetheless, both methods perform well and can be considered suitable to the problem of AS-level traffic modelling.

## 5.5 Summary

As illustrated in the previous section, both methods are quite capable of predicting data over a short-to-mid-term horizon. From our perspective, this is quite pleasing. The *time-series* DHR method is very well-suited to the problem, and as mentioned has been applied to similar types of systems in the past. Therefore, that quite a simple kernel-based method can be used to achieve comparative levels of performance illustrates how capable such approaches are.

Furthermore, what is very appealing about the kernel-based approach is how few user-choices need to be made in order to estimate a reasonably accurate model - meaning replication of the

| Site | HINET | LUCIX | NETFLIX |
|---|---|---|---|
| AS Number | 3460 | 49624 | 2906 |
| Flow Direction | Outgoing | Incoming | Incoming |
| $\mathcal{M}_{\text{DHR}}$ | | | |
| AR Order | 52 | 95 | 73 |
| TVP | Trig. | Trig. | Trig. |
| FIT (%) | 80.22 | 83.95 | 85.71 |
| $\mathcal{M}_{\text{SID}}$ | | | |
| Kernel Width ($\sigma$) | $[\,14.2 \quad 2.8\,]^\top$ | $[\,14.2 \quad 2.8\,]^\top$ | $[\,14.2 \quad 2.8\,]^\top$ |
| Reg. Strength ($\lambda$) | 0 | 2.07 | 0.37 |
| FIT (%) | 94.89 | 93.01 | 92.12 |

TABLE 5.1 – Results of $\mathcal{M}_{\text{DHR}}$ and $\mathcal{M}_{\text{SID}}$

method onto other systems in the network is straightforward.

As mentioned, this method is currently in the final stages of development prior to online deployment, as a bespoke package for AS-level modelling and fault-detection (the ANODE framework), which has been developed by colleagues at Post. With this package now implemented, it is hoped that data can now be used to further improve the results, with several different diretions envisaged over the coming months:

- **Structural optimisation:** the linear + nonlinear structure implemented in this chapter in fact corresponds to a very simple Hammerstein structure. Whilst this has already been used to obtain good results, it would nonetheless be interesting to investigate how other types of nonlinear structures perform in order to improve the accuracy of the models.
- **Complexity tuning:** similarly, in this chapter a gradient penalty was used. As illustrated in section 4.2, this offers a way of applying a soft smoothness constraint capable of distinguishing between smooth and nonsmooth behaviour. However, it would also be interesting to investigate how other penalties could be used to improve the overall performance of the model - as discussed in section 4.4.
- **Modelling time-varying behaviour:** in their current form, one advantage of the DHR method over kernel method used is its ability to model time-varying behaviour (as opposed to time-periodic behaviour). Whilst this is not too restrictive for prediction over a relatively short-term window (< 3 months), for longer-term predictions based on a longer training period (e.g. 6 months+), the ability to model time-varying behaviour would be very useful. There are potentially many ways of doing this. Currently, the ANODE framework uses an offline sliding window method, computing predictions for the coming week based on the past 2-4 weeks of observations. Alternatively, a regularisation/forgetting-factor approach could be interesting, or even the consideration of a time-varying kernel function. For example, the stable Gaussian RBF kernel discussed in the literature (e.g. [Darwish et al., 2015]) could be used to prioritise newer information in the model.
- **Computational optimisation:** at present, forecasting is based on weekly offline computation of each model individually. Whilst this is feasible and not problematic, it is nonetheless time-consuming as around 30 AS need to be modelled. It would be interesting to see how this process could be optimised, by using some sort of recursive/online computation of the models. This would reduce computational time and server load, and also facilitate an increased frequency of recomputation.
- **Adaptation to different applications:** as noted, it is also desirable to have a model

that can be adapted to different applications. In fact, in the current setup scope already exists for this. In the previous section, models were evaluated in full-simulation over a period of 4 weeks. However, changing the prediction horizon would facilitate the detection of different phenomena. For example, a one-step ahead predictor could be used to detect network faults. A ten-fifteen minute horizon could be used to detect attacks on the network. A three-six hour window could help detect short duration events, and a one-day horizon could be used to detect events such as holidays (Christmas, Easter, Bank Holidays etc). In this way, the same model can be re-adapted to meet the needs of the ISP with minimal effort.

# 6

# Conclusions

## 6.1 Summary

In this thesis we have looked at several important questions in nonlinear identification, notably those of:

- How to define a nonlinear model, and
- How to control the properties of a nonlinear model.

These questions were incorporated into a discussion on the use of kernel methods - a class of nonparametric methods - in nonlinear system identification. This discussion centered on two distinct but complementary approaches in the literature, namely:

- *Reproducing Kernel Hilbert Spaces* (RKHS) methods and
- *Sobolev Spaces* methods.

In both cases, a kernel function is used to define an associated model class, with a regularisation term used to control the bias-variance trade-off of the estimated model. However, in practice the way in which these two methods operate is very different.

- In the RKHS, the model properties are almost exclusively tuned through the kernel function, with the regularisation acting primarily to ensure well-posedness of the solution.
- In the Sobolev Space, the kernel function is implicit in the definition of the norm of the space and usually defines an *a priori* flexible model class, with the regularisation term now acting both on the numerical stability and the properties of the model.

These two methods of acting on the model properties were referred to as *hard* and *soft* constraints respectively, to denote the fact that placing constraints on the model through the kernel function places strict limits on the model properties, whereas placing *soft* constraints through a regularisation term places a *top-down* constraint on the model properties, which allows the model to adhere more to the data where informative, and more to the *a priori* placed by the constraint otherwise.

Both are very appealing. In particular, placing hard constraints through the kernel allows for structural chioces to be encoded into the model definition. And, placing soft constraints through a regularisation term offers a way to incorporate properties such as smoothness into the model as desired - in certain cases with reduced bias incorporated into the model. Furthermore, in certain examples these two methods can actually achieve very similar levels of control over the model properties - despite acting in very different ways.

To this end, this thesis examined two methods of penalising derivatives in an RKHS, without constraining the choice of the kernel function. Such methods allow both hard and soft constraints to applied, and offer greater freedom in the types of soft constraints that can be formulated: as they no longer depend upon finding the optimal *Sobolev* kernel. Whilst both methods rely on some form approximation - either in the formulation of the cost-function (the *indirect* approach) or in the definition of a representer (the *direct* approach) - they can nonetheless achieve similar levels of performance and control over the model properties when compared with the optimal Sobolev approach from the literature.

To illustrate how such methods could be used in practice, several examples were presented - comparing the traditional RKHS approach with the direct method proposed in this thesis. In each case, the examples were chosen to highlight a different aspect of the methods:

- The first example illustated the advantage of applying soft constraints on the model smoothness, using a regularisation over the functional gradient to differentiate between smooth and nonsmooth behaviour in a simple, static example.
- The second example illustrated how RKHS methods can be applied to dynamical control-relevant model structures, using *linear parameter-varying* (LPV) models as an example. Furthermore, once the standard problem is formulated, extension to the derivative case is straightforward: requiring only determination of the relevant derivative kernel.
- The final simulation example illustrated how formulating penalties in an RKHS can allow for control over model properties in a way otherwise difficult to achieve using either a Sobolev or a standard RKHS approach. This potentially offers the user many new tools, but in this case we focused on how a separability penalty can be used in conjunction with a functional-norm penalty, to allow an additional manner of tuning the model properties.
- And a real-data example was also presented: modelling *autonomous system* (AS) level web traffic. In this case, it was shown that a kernel-based approach offers a powerful, flexible and user-friendly framework for modelling - with a derivative constraint allowing the optimisation problem to be further simplified by removing the need to optimise over the kernel hyperparameters.

## 6.2 Conclusions

Kernel methods offer a very attractive framework for nonlinear identification, as they are both capable of being applied to different types of problems and capable of modelling many different types of nonlinearities. Additionally, incorporating derivative penalties into the optimisation criterion through a regularisation term offers another way of controlling model properties, which is appealing in many different circumstances.

The theoretically-exact manner of penalising derivatives requires formulation of the problem in a Sobolev Space. For known problems, this is then an easy way of controlling properties, such as smoothness. However, for 'non-standard' problems this is less trivial, as it requires that the corresponding *Green's Function* problem can be both formulated and solved. In some cases, exact solutions may not be possible to find, and even when they do exist it may not be straightforward to find them.

However, such penalties can also be formulated in an RKHS directly. This provides a framework consistent with traditional RKHS methods, in that the user is allowed freedom in the choice of kernel function, and as shown can achieve similar results to the theoretically optimal methods.

Care should be taken when using such methods. In the case of the *direct* method, an approximation in the representer is necessary to avoid constraining the choice of kernel function. This can be partly resolved by adding kernels to the definition of the representer, or in the case of RBF kernels, determining the width of the kernel with respect to the density of kernels of the input space. Alternatively, the kernel function can be included in the optimisation problem as would usually be done. In the case of the *indirect* method, the same problems do not arise in the formulation - from a theoretical point-of-view. However in practice, care should be taken in the kernel selection in the same way - to ensure that the model properties are globally controlled, and not only point-wise constrained.

Such methods are potentially useful in many different types of problems - such as structural detection, structural approximation and complexity tuning - and offer an application-driven framework for nonlinear identification, in which the user can think about what properties are desirable in a model and enforce them as much as required.

## 6.3 Future Work

This thesis has focused on the examination of problems already discussed in the literature (controlling smoothness) and problems that extend from the literature (enforcing separability), with the aim of understanding how such methods work and how they can be interesting in practically-oriented examples. However, the scope of such methods is by no means limited to the problems discussed in this thesis.

Penalising model properties using derivative regularisation not only gives us a way of controlling properties which is distinct from the way in which they can be controlled through the kernel function - as in the separability examples presented in section 4.4 - but also a way of controlling properties that cannot be controlled through the kernel at all.

For example, throughout the thesis we have focused on the case of the Gaussian RBF kernel. The Gaussian kernel is a very useful example. It is both well-understood and widely-used in practice, and acts as a kernelised 'blank canvas' - in that it can handle many different types of problems. However, other choices of kernels would of course also be interesting to investigate. For example, polynomial and thin-plate spline kernels - that are naturally restricted in terms of their continuity - offer the possibility to enforce properties such as multiplicative separability, in a manner akin to additive separability in Chapter 4.

Derivative methods could also offer a novel way of formulating grey-box nonlinear identification problems: in which the kernel function is used to place structural constraints, and the regularisation term used to describe the physical behaviour of the dynamical system - as discussed in [Ramsay and Silverman, 1997].

In both cases, derivative penalties could be a useful way of either improving the accuracy of a model or gaining physical insight from the system - depending on the scenario and the requirements of the user.

# Bibliographie


[Adams, 1975] Adams, R. (1975). Sobolev spaces–new york. *San Francisco. Lodon : Academic Press*, 1 :975.

[Ager et al., 2012] Ager, B., Chatzis, B., Feldmann, A., Sarrar, N., Uhlig, S., and Willinger, W. (2012). Anatomy of a large european ixp. In *SIGCOMM'12*, pages 163–174, Helsinki, Finland.

[Akhiezer and Glazman, 1963] Akhiezer, N. and Glazman, I. (1963). *Theory of Linear Operators in Hilbert Space*. Ungar, New York.

[Alarcon-Aquino and Barria, 2006] Alarcon-Aquino, V. and Barria, J. (2006). Multiresolution fir neural-network-based learning algorithm applied to network traffic prediction. *IEEE Transactions on Systems, Man, and Cybernetics — Part C : Applications and Reviews*, 36(2) :208–220.

[Alkhoury et al., 2017] Alkhoury, Z., Petreczky, M., and Mercère, G. (2017). Identifiability of affine linear parameter-varying models. *Automatica*, 80 :62–74.

[Arfken, 1985] Arfken, G. (1985). The method of steepest descents. *Mathematical methods for physicists*, 3 :428–436.

[Aronszajn, 1950] Aronszajn, N. (1950). Theory of reproducing kernels. *Transactions of the American Mathematical Society*, 68(3) :337–404.

[Åström and Eykhoff, 1971] Åström, K. J. and Eykhoff, P. (1971). System identification—a survey. *Automatica*, 7(2) :123–162.

[Awduche et al., 2002] Awduche, D., Chiu, A., Elwalid, A., Widjaja, I., and Xiao, X. (2002). Overview and principles of internet traffic engineering. RFC 3272, RFC Editor.

[Bai and Liu, 2007] Bai, E. and Liu, Y. (2007). Recursive direct weight optimization in nonlinear system identification : a minimal probability approach. *IEEE Transactions on Automatic Control*, 52 :1218–1231.

[Bai, 2005] Bai, E.-W. (2005). Identification of an additive nfir system and its applications in generalized hammerstein models. In *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC'05. 44th IEEE Conference on*, pages 6406–6411. IEEE.

[Bamieh and Giarré, 2002] Bamieh, B. A. and Giarré, L. (2002). Identification of linear parameter-varying models. *International Journal of Robust Nonlinear Control*, 12(9) :841–853.

[Barakat et al., 2002] Barakat, C., Thiran, P., Iannaccone, G., Diot, C., and Owezarski, P. (2002). A flow-based model for internet backbone traffic. In *Proceedings of the 2Nd ACM SIGCOMM Workshop on Internet Measurment*, IMW '02, pages 35–47, New York, NY, USA. ACM.

[Bates et al., 2016] Bates, T., Smith, P., and Huston, G. (2016). CIDR report for 9 may 17. Technical report, CIDR.

[Bellman, 1957] Bellman, R. (1957). Dynamic programming princeton university press princeton. *New Jersey Google Scholar*.

[Berlinet and Thomas-Agnan, 2004] Berlinet, A. and Thomas-Agnan, C. (2004). *Reproducing Kernels in Probability and Statistics*. Springer.

[Bhujwalla et al., 2017a] Bhujwalla, Y., Grandemange, Q., Gilson, M., Laurain, V., and Gnaedinger, E. (2017a). How we spend our time online : Predicting internet traffic using system identification. In *Proceedings of IFAC World Congress*, Toulouse, France.

[Bhujwalla et al., 2016a] Bhujwalla, Y., Laurain, V., and Gilson, M. (2016a). The impact of smoothness on model class selection in nonlinear system identification : An application of derivatives in the rkhs. In *Proceedings of American Control Conference (ACC)*, Boston, Massachusetts, USA.

[Bhujwalla et al., 2016b] Bhujwalla, Y., Laurain, V., and Gilson, M. (2016b). An RKHS approach to systematic kernel selection in nonlinear system identification. In *Proceedings of $55^{th}$ IEEE Conference for Decision and Control (CDC)*, Las Vegas, Nevada, USA.

[Bhujwalla et al., 2017b] Bhujwalla, Y., Laurain, V., Gilson, M., and Gnaedinger, E. (2017b). An RKHS approach to controlling smoothness in nonparametric LPV-IO identification. In *Proceedings of IFAC World Congress*, Toulouse, France.

[Billings, 1980] Billings, S. A. (1980). Identification of nonlinear systems - a survey. *IEEE Proceedings of Control Theory*, 127 :272–285.

[Billings, 2013] Billings, S. A. (2013). *Nonlinear system identification : NARMAX methods in the time, frequency, and spatio-temporal domains*. John Wiley & Sons.

[Billings and Chen, 1989] Billings, S. A. and Chen, S. (1989). Identification of non-linear rational systems using a prediction-error estimation algorithm. *International Journal of Systems Science*, 20 :467–494.

[Bombois et al., 2006] Bombois, X., Scorletti, G., Gevers, M., Van den Hof, P. M., and Hildebrand, R. (2006). Least costly identification experiment for control. *Automatica*, 42(10) :1651–1662.

[Box, 1976] Box, G. (1976). Science and statistics. *Journal of the American Statistical Association*, 71 :791–799.

[Box et al., 1970] Box, G. E., Jenkins, G. M., and Reinsel, G. (1970). Forecasting and control. *Time Series Analysis*, 3 :75.

[Brunet, 1989] Brunet, R. (1989). *Les Villes européennes, Rapport pour la DATAR*. La Documentation Française, Paris.

[Cerrone et al., 2017] Cerrone, V., Fadda, E., and Regruto, D. (2017). A robust optimization approach to kernel-based nonparametric error-in-variables identification in the presence of bounded noise. In *Proceedings of the 2017 American Control Conference ACC*, Seattle, USA.

[Chen and Ljung, 2015] Chen, T. and Ljung, L. (2015). On kernel structure for regularized system identification (i) and (ii). In *IFAC SYSID*, Beijing, China.

[Chen et al., 2012a] Chen, T., Ljung, L., Andersen, M., Chiuso, A., Carli, F., and Pillonetto., G. (2012a). Sparse multiple kernels for impulse response estimation with majorization minimization algorithms. In *Proceedings of the 50th IEEE Conference on Decision and Control*, pages 1500–1505, Hawaii, USA.

[Chen et al., 2011] Chen, T., Ohlsson, H., Goodwin, G. C., and Ljung, L. (2011). Kernel selection in linear system identification part ii : A classical perspective. In *Decision and Control and*

*European Control Conference (CDC-ECC), 2011 50th IEEE Conference on*, pages 4326–4331. IEEE.

[Chen et al., 2012b] Chen, T., Ohlsson, H., and Ljung, L. (2012b). On the estimation of transfer functions, regularizations and gaussian processes - revisited. *Automatica*, 48(8) :1525–1535.

[Chiuso, 2016] Chiuso, A. (2016). Regularization and bayesian learning in dynamical systems : Past, present and future. *Annual Reviews in Control*, 41 :24–38.

[Cisco, 2015] Cisco (2015). The zettabyte era : Trends and analysis. White Paper.

[Claise, 2004] Claise, B. (2004). Cisco systems netflow services export version 9. RFC 3954, RFC Editor.

[Claise et al., 2013] Claise, B., Trammell, B., and Aitken, P. (2013). Specification of the ip flow information export (IPFIX) protocol for the exchange of flow information. RFC 7011, RFC Editor.

[Clements and Hendry, 2001] Clements, M. and Hendry, D. (2001). *A Companion to Economic Forecasting*. Oxford : Blackwells.

[Cortes and Vapnik, 1995] Cortes, C. and Vapnik, V. (1995). Support vector machine. *Machine learning*, 20(3) :273–297.

[Cortez et al., 2012] Cortez, P., Rio, M., Rocha, M., and Sousa, P. (2012). Multi-scale internet traffic forecasting using neural networks and time series methods. *Expert Systems*, 2 :143–155.

[Cucker and Zhou, 2007] Cucker, F. and Zhou, D. (2007). *Learning Theory : An Approximation Theory Viewpoint*. Cambridge University Press.

[Dalchau, 2012] Dalchau, N. (2012). Understanding biological timing using mechanistic and black-box models. *New Phytologist*, 193(4) :852–858.

[Darwish et al., 2015] Darwish, M., Cox, P., Pillonetto, G., and Tóth, R. (2015). Bayesian identification of LPV Box-Jenkins models. In *Proc. of the 54th IEEE Conference on Decision and Control*, pages 66–71, Osaka, Japan.

[De Brabanter et al., 2011] De Brabanter, K., De Brabanter, J., Suykens, J. A., and De Moor, B. (2011). Approximate confidence and prediction intervals for least squares support vector regression. *IEEE Transactions on Neural Networks*, 22(1) :110–120.

[Ding et al., 1995] Ding, X., Canu, S., and Denoeux, T. (1995). Neural network based models for forecasting. In *Proceedings of Applied Decision Technologies Conference (ADT'95)*, pages 243—252, Uxbridge, UK.

[Doumpos et al., 2007] Doumpos, M., Zopounidis, C., and Golfinopoulou, V. (2007). Additive support vector machines for pattern classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 37(3) :540–550.

[Dreesen et al., 2015] Dreesen, P., Schoukens, M., Tiels, K., and Schoukens, J. (2015). Decoupling static nonlinearities in a parallel wiener-hammerstein system : A first-order approach. In *Instrumentation and Measurement Technology Conference (I2MTC), 2015 IEEE International*, pages 987–992. IEEE.

[Duchon, 1977] Duchon, J. (1977). Splines minimizing rotation-invariant semi-norms in sobolev spaces. *Constructive theory of functions of several variables*, pages 85–100.

[Duijkers et al., 2014] Duijkers, R., Tóth, R., Piga, D., and Laurain, V. (2014). Shrinking complexity of scheduling dependencies in LS-SVM based LPV system identification. In *Proceedings of the 53rd IEEE Conference on Decision and Control*, pages 2561 – 2566, Los Angeles, California, USA.

[Duvenaud, 2014] Duvenaud, D. (2014). *Automatic Model Construction with Gaussian Processes*. PhD thesis, Computational and Biological Learning Laboratory, University of Cambridge.

[Duvenaud et al., 2013] Duvenaud, D., Lloyd, J. R., Grosse, R., Tenenbaum, J. B., and Ghahramani, Z. (2013). Structure discovery in nonparametric regression through compositional kernel search. *arXiv preprint arXiv :1302.4922*.

[Duvenaud et al., 2011] Duvenaud, D. K., Nickisch, H., and Rasmussen, C. E. (2011). Additive gaussian processes. In *Advances in neural information processing systems*, pages 226–234.

[Espinoza et al., 2005] Espinoza, M., Suykens, J. A., and De Moor, B. (2005). Kernel based partially linear models and nonlinear identification. *IEEE Transactions on Automatic Control*, 50(10) :1602–1606.

[Faratin et al., 2008] Faratin, P., Clarck, D. D., Bauer, S., Lehr, W., Gilmore, P. W., and Berger, A. (2008). The growing complexity of internet interconnection. *Communications and Strategies*, 72 :51.

[Feldmann et al., 2001] Feldmann, A., Greenberg, A., Lund, C., Reingold, N., Rexford, J., and True, F. (2001). Deriving traffic demands for operational ip networks : Methodology and experience. *IEEE/ACM Trans. Netw.*, 9(3) :265–280.

[Garnier and Wang, 2008] Garnier, H. and Wang, L. (2008). *Continuous-time Models from Sampled Data*. Springer-Verlag.

[Gevers, 2006] Gevers, M. (2006). A personal view of the development of system identification : A 30-year journey through an exciting field. *IEEE Control Systems*, 26(6) :93–105.

[Gevers et al., 2008] Gevers, M., Bazanella, A., and Miskovi, L. (2008). Informative data : how to get just sufficiently rich ? In *Decision and Control, 2008. CDC 2008. 47th IEEE Conference on*, pages 1962–1967. IEEE.

[Gilson et al., 2012] Gilson, M., Laurain, V., Garnier, H., Payraudeau, S., and Grégoire, C. (2012). A new data-based modelling method for identifying parsimonious nonlinear rainfall/flow models. *Journal Européen des Systèmes Automatisés (JESA)*, 46(6-7) :633–647.

[Gilson and Van Den Hof, 2005] Gilson, M. and Van Den Hof, P. (2005). Instrumental variable methods for closed-loop system identification. *Automatica*, 41(2) :241–249.

[Giri and Bai, 2010] Giri, F. and Bai, E.-W. (2010). *Block-oriented nonlinear system identification*, volume 1. Springer.

[Goethals et al., 2005a] Goethals, I., Pelckmans, K., Suykens, J. A., and De Moor, B. (2005a). Identification of mimo hammerstein models using least squares support vector machines. *Automatica*, 41(7) :1263–1272.

[Goethals et al., 2005b] Goethals, I., Pelckmans, K., Suykens, J. A., and De Moor, B. (2005b). Subspace identification of hammerstein systems using least squares support vector machines. *IEEE Transactions on Automatic Control*, 50(10) :1509–1519.

[Gohnen and Alpaydin, 2011] Gohnen, M. and Alpaydin, E. (2011). Multiple kernel learning algorithms. *Journal of Machine Learning Research*, 12 :2211–2268.

[Golabi et al., 2014] Golabi, A., Meskin, N., Tóth, R., and Mohammadpour, M. (2014). A bayesian approach for estimation of LPV linear-regression models. In *Proc. of the 53rd IEEE Conference on Decision and Control*, pages 2555–2560, Los Angeles, CA, USA.

[Gonzàlez and Moral, 1997] Gonzàlez, P. and Moral, P. (1997). Comments on 'an analysis of the international tourism demands in spain'. *International Journal of Forecasting*, 13 :551–556.

[Grandemange et al., 2017a] Grandemange, Q., Bhujwalla, Y., Gilson, M., Ferveur, O., and Gnaedinger, E. (2017a). Analysing and modelling a network as-level traffic. In *20th IFAC World Congress, IFAC 2017*.

[Grandemange et al., 2017b] Grandemange, Q., Bhujwalla, Y., Gilson, M., and Gnaedinger, E. (2017b). An AS-level approach to network traffic analysis and modelling. In *Proceedings of IEEE International Conference on Communications (ICC)*, Paris, France.

[Grandemange et al., 2017c] Grandemange, Q., Ferveur, O., Gilson, M., and Gnaedinger, E. (2017c). A live network AS-level traffic characterization. In *Proceedings of IEEE International Conference on Computing, Networking and Communications (ICNC)*, Silicon Valley, USA.

[Hadamard, 1902] Hadamard, J. (1902). *Sur les problèmes aux dérivées partielles et leur signification physique*. Princeton University Bulletin.

[Hall et al., 2012] Hall, J., Rasmussen, C., and Maciejowski, J. (2012). Modelling and control of nonlinear systems using gaussian processes with partial model information. In *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*, pages 5266–5271. IEEE.

[Halmos, 1957] Halmos, P. (1957). *Introduction to Hilbert Space and the Theory of Spectral Multiplicity*. Chelsea, New York.

[Hasselmann et al., 1963] Hasselmann, K., Munk, W., and MacDonald, G. (1963). *Time Series Analysis*. Wiley.

[Hastie and Tibshirani, 1990] Hastie, T. and Tibshirani, R. (1990). *Generalized additive models*. Wiley Online Library.

[Hofstede et al., 2014a] Hofstede, R., Celeda, P., Trammell, B., Drago, I., Sadre, R., Sperotto, A., and Pras, A. (2014a). Flow monitoring explained : From packet capture to data analysis with netflow and ipfix. *IEEE Communications Surveys Tutorials*, 16(4) :2037–2064.

[Hofstede et al., 2014b] Hofstede, R., Celeda, P., Trammell, B., Drago, I., Sadre, R., Sperotto, A., and Pras, A. (2014b). Flow monitoring explained : From packet capture to data analysis with netflow and ipfix. *Flow Monitoring Explained : From Packet Capture to Data Analysis With NetFlow and IPFIX*, 16(4) :2037–2064.

[Hong et al., 2008] Hong, X., Mitchell, R. J., Chen, S., Harris, C. J., Li, K., and Irwin, G. W. (2008). Model selection approaches for non-linear system identification : a review. *International journal of systems science*, 39(10) :925–946.

[Hsu et al., 2008] Hsu, K., Vincent, T. L., and Poolla, K. (2008). Nonparametric methods for the identification of linear parameter varying systems. In *Proc. of the Int. Symposium on Computer-Aided Control System Design*, pages 846–851, San Antonio, Texas, USA.

[Jakeman and Young, 1984] Jakeman, A. and Young, P. (1984). Recursive filtering and the inversion of ill-posed causal problems. *Utilitas Mathematica*, 35 :351–376.

[James et al., 2014] James, G., Witten, D., and Hastie, T. (2014). An introduction to statistical learning : With applications in r.

[Jiang et al., 2010] Jiang, H., Ge, Z., Jin, S., and Wang, J. (2010). Network prefix-level traffic profiling : Characterizing, modeling, and evaluation. *Computer Networks*, 54(18) :3327 – 3340.

[Judge et al., 1982] Judge, G. G., Hill, R. C., Griffiths, W., Lutkepohl, H., and Lee, T. C. (1982). *Introduction to the Theory and Practice of Econometrics*. New York New York John Wiley and Sons 1982.

[Judistky et al., 1995] Judistky, A., Hjalmarsson, H., Beneviste, A., Deylon, B., Ljung, L., Sjöberg, J., and Zhang, Q. (1995). Nonlinear black-box modelling in system identification : Mathematical foundations. *Automatica*, 31(12) :1725–1752.

[Kalman, 1960] Kalman, R. (1960). A new approach to linear filtering and prediction problems. *ASME Transactions Journal of Basic Engineering*, 83 :95–108.

[Kimeldorf and Wahba, 1971] Kimeldorf, G. and Wahba, G. (1971). Some results on tchebycheffian spline functions. *Journal of Mathematical Analysis and Applications*, 33(1) :82–95.

[Kimenta, 1986] Kimenta, J. (1986). *Elements of econometrics*. Ann Arbor : MI : University of Michigan Press.

[Kondor et al., 2005] Kondor, R., Csányi, G., Ahnert, S., and Jebara, T. (2005). Multi facet learning in hilbert spaces. *Internal Technical Report for Columbia University*.

[Krige, 1966] Krige, D. G. (1966). A study of gold and uranium distribution patterns in the klerksdorp gold field. *Geoexploration*, 4(1) :43–53.

[Kulkarni et al., 2014] Kulkarni, V., Stan, G.-B., and Raman, K. (2014). *A systems theoretic approach to systems and synthetic biology I : models and system characterizations*, volume 1. Springer.

[Labovitz et al., 2010a] Labovitz, C., Iekel-Johnson, S., McPherson, D., Oberheide, J., and Jahanian, F. (2010a). Internet inter-domain traffic. In *SIGCOMM'10*, pages 75–86, New Delhi, India.

[Labovitz et al., 2010b] Labovitz, C., Iekel-Johnson, S., McPherson, D., Oberheide, J., and Jahanian, F. (2010b). Internet inter-domain traffic. In *Proceedings of ACM Special Interest Group on Data Communication (SIGCOMM)*, New Delhi, India.

[Lataire and Chen, 2016] Lataire, J. and Chen, T. (2016). Transfer function and transient estimation by gaussian process regression in the frequency domain. *Automatica*, 72 :217–229.

[Lauer et al., 2012] Lauer, F., Le, V., and Bloch, G. (2012). Learning smooth models of nonsmooth functions via convex optimization. In *Proceedings of the IEEE Int. Workshop on Machine Learning for Signal Processing*, Santander, Spain.

[Laurain et al., 2014] Laurain, V., Gilson, M., and Benoît, M. (2014). Data-driven modeling for water resource quality over long term trends. In *7th International Congress on Environmental Modelling and Software, iEMSs 2014*.

[Laurain et al., 2010] Laurain, V., Gilson, M., Tóth, R., and Garnier, H. (2010). Refined instrumental variable methods for identification of LPV Box–Jenkins models. *Automatica*, 46(6) :959–967.

[Laurain et al., 2015] Laurain, V., Tóth, R., Piga, D., and Zheng, W. (2015). An instrumental least squares support vector machine for nonlinear systems identification. *Automatica*, 54 :340–347.

[Laurain et al., 2012] Laurain, V., Tóth, R., Zheng, W., and Gilson, M. (2012). Nonparametric identification of LPV models under general noise conditions, an LS-SVM based approach. In *Proc. of the the 16th IFAC Symposium on System Identification*, pages 1761–1766, Brussels, Belgium.

[Lin et al., 1996] Lin, T., Horne, B., Tino, P., and Giles, C. (1996). Learning long-term dependencies in NARX recurrent neural networks. *IEEE Transactions on Automatic Control*, 7 :1329–1338.

[Ljung, 1999] Ljung, L. (1999). *System Identification, theory for the user*. Prentice Hall.

[Ljung, 2010] Ljung, L. (2010). Perspectives on system identification. *Annual Reviews in Control*, 34(1) :1–12.

[Lovera and Mercere, 2007] Lovera, M. and Mercere, G. (2007). Identification for gain-scheduling : a balanced subspace approach. In *American Control Conference, 2007. ACC'07*, pages 858–863. IEEE.

[MacKay, 2003] MacKay, D. (2003). *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press.

[MacKay, 2009] MacKay, D. (2009). *Sustainable Energy - without the hot air*. UIT, Cambridge.

[MacKay, 1998] MacKay, D. J. (1998). Introduction to gaussian processes. *NATO ASI Series F Computer and Systems Sciences*, 168 :133–166.

[Manton and Amblard, 201] Manton, J. and Amblard, P. (201). A primer on reproducing kernel hilbert spaces. *Foundations and Trends in Signal Processing*, 8 :1–133.

[Marconato and Schoukens, 2009] Marconato, A. and Schoukens, J. (2009). Identification of wiener-hammerstein benchmark data by means of support vector machines. *IFAC Proceedings Volumes*, 42(10) :816–819.

[Marconato et al., 2016] Marconato, A., Schoukens, M., and Schoukens, J. (2016). Filter-based regularisation for impulse response modelling. *IET Control Theory & Applications*, 11(2) :194–204.

[Martin, 2012] Martin, C. (2012). Kernels part 1 : What is an rbf kernel really ? Technical report.

[Mehrkanoon et al., 2012] Mehrkanoon, S., Falck, T., and Suykens, J. A. (2012). Approximate solutions to ordinary differential equations using least squares support vector machines. *IEEE transactions on neural networks and learning systems*, 23(9) :1356–1367.

[Mehrkanoon and Suykens, 2012] Mehrkanoon, S. and Suykens, J. A. (2012). Ls-svm approximate solution to linear time varying descriptor systems. *Automatica*, 48(10) :2502–2511.

[Meinguet, 1979] Meinguet, J. (1979). Multivariate interpolation at arbitrary points made simple. *Zeitschrift für angewandte Mathematik und Physik (ZAMP)*, 30(2) :292–304.

[Mercer, 1909] Mercer, J. (1909). Functions of positive and negative type, and their connection with the theory of integral equations. *Philosophical Transactions of the Royal Society of London, Series A*, 209 :415–446.

[Minh, 2010] Minh, H. (2010). Some properties of gaussian reproducing kernel hilbert spaces and their implications for function approximation and learning theory. *Constructive Approximation*, 32 :307–338.

[Moore, 1916] Moore, E. H. (1916). On properly positive hermitian matrices. *Bull. Amer. Math. Soc*, 23(59) :66–67.

[Mu et al., 2017a] Mu, B., Chen, T., and Ljung, L. (2017a). On asymptotic properties of hyperparameter estimators for kernel-based regularization methods. *arXiv preprint arXiv :1707.00407*.

[Mu et al., 2017b] Mu, B., Zheng, W., , and Bai, E. (2017b). Variable selection and identification of high-dimensional nonparametric additive nonlinear systems. *IEEE Transactions on Automatic Control*, 62(5) :2254–2269.

[Mukherjee and Zhou, 2006] Mukherjee, S. and Zhou, D.-X. (2006). Learning coordinate covariances via gradients. *Journal of Machine Learning Research*, 7(Mar) :519–549.

[Nelles, 2001] Nelles, O. (2001). *Nonlinear System Identification : From Classical Approaches to Neural Networks and Fuzzy Models*. Springer-Verlag, Berlin.

[Newton, 2002] Newton, H. (2002). A conversation with emmanuel parzen. *Statistical Science*, 17(3) :357–378.

[Nosedal-Sanchez et al., 2012] Nosedal-Sanchez, A., Storlie, C., Lee, T., and Christensen, R. (2012). Reproducing kernel hilbert spaces for penalized regression : A tutorial. *The American Statistician*, 66(1) :50–60.

[O'Sullivan, 1986] O'Sullivan, F. (1986). A statistical perspective on ill-posed inverse problems. *Statistical science*, pages 502–518.

[Parzen, 1960] Parzen, E. (1960). *Modern Probability Theory and its Application*. Wiley.

[Piga and Tóth, 2013] Piga, D. and Tóth, R. (2013). LPV model order selection in an LS-SVM setting. In *Proc. of the 52nd IEEE Conference on Decision and Control*, pages 4128–4133, Florence, Italy.

[Pillonetto et al., 2011] Pillonetto, Quang, G., and Chiuso, A. (2011). A new kernel-based approach for nonlinear system identification. *IEEE Transactions on Automatic Control*, 56(12) :2825–2840.

[Pillonetto, 2008] Pillonetto, G. (2008). Identification of time-varying systems in reproducing kernel hilbert spaces. *IEEE Transactions on Automatic Control*, 53(9) :2202–2209.

[Pillonetto and De Nicolao, 2010] Pillonetto, G. and De Nicolao, G. (2010). A new kernel-based approach for linear system identification. *Automatica*, 46(1) :81–93.

[Pillonetto and De Nicolao, 2011] Pillonetto, G. and De Nicolao, G. (2011). Kernel selection in linear system identification part i : A gaussian process perspective. In *Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on*, pages 4318–4325. IEEE.

[Pillonetto et al., 2014] Pillonetto, G., Dinuzzo, F., Chen, T., Nicolao, G., and Ljung, L. (2014). Kernel methods in system identification, machine learning and function estimation : A survey. *Automatica*, 50(3) :657–682.

[Pintelon and Schoukens, 2012] Pintelon, R. and Schoukens, J. (2012). *System identification, a frequency domain approach*. IEEE press.

[Poggio and Girosi, 1990] Poggio, T. and Girosi, F. (1990). Networks for approximation and learning. *Proceedings of the IEEE*, 78(9) :1481–1497.

[Ramsay and Silverman, 1997] Ramsay, J. and Silverman, T. (1997). *Functional Data Analysis*. Springer.

[Ramsay and Silverman, 2002] Ramsay, J. and Silverman, T. (2002). *Applied Functional Data Analysis : Methods and Case Studies*. Springer.

[Rasmussen and Williams, 2006] Rasmussen, C. and Williams, C. (2006). *Gaussian Processes for Machine Learning*. The MIT Press.

[Rasmussen et al., 2016] Rasmussen, C., Williams, C., and Nickisch, H. (2016). Documentation for GPML Matlab code version 4.0.

[Rasmussen and Ghahramani, 2001] Rasmussen, C. E. and Ghahramani, Z. (2001). Occam's razor. In *Advances in neural information processing systems*, pages 294–300.

[Riesz and Szőkefalvi-Nagy, 1953] Riesz, F. and Szőkefalvi-Nagy, B. (1953). *Leçons d'analyse fonctionelle*. Akadémiai Kiadó Budapest.

[Risuleo et al., 2015a] Risuleo, R., Bottegal, G., and Hjarlmarsson, H. (2015a). A kernel-based approach to hammerstein system identication. In *Proceedings of IFAC Symposium on System Identification*, volume 48, pages 1011–1016, Beijing, China.

[Risuleo et al., 2015b] Risuleo, R., Molinari, M., Bottegal, G., Hjarlmarsson, H., and Johansson, K. (2015b). A benchmark for data-based office modeling : challenges related to co2 dynamics. In *Proceedings of IFAC Symposium on System Identification*, volume 48, pages 1256–1261, Beijing, China.

[Risuleo, 2016] Risuleo, R. S. (2016). *System identification with input uncertainties : an EM kernel-based approach*. PhD thesis, KTH Royal Institute of Technology.

[Roll et al., 2005] Roll, J., Nazin, A., and Ljung, L. (2005). Nonlinear system identification via direct weight optimization. *Automatica*, 41 :475–490.

[Romeny, 2008] Romeny, B. M. H. (2008). *Front-end vision and multi-scale image analysis : multi-scale computer vision theory and applications, written in mathematica*, volume 27. Springer Science & Business Media.

[Rosasco et al., 2010] Rosasco, L., Santoro, M., Mosci, S., Verri, A., and Villa, S. (2010). A regularization approach to nonlinear variable selection. *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics*, 9 :653——660.

[Rosasco et al., 2014] Rosasco, L., Villa, S., Mosci, S., Santoro, M., and Verri, A. (2014). Nonparametric sparsity and regularization. *Journal of Machine Learning Research*, 14 :1665–1714.

[Roughan et al., 2011] Roughan, M., Willinger, W., Maennel, O., Perouli, D., and Bush, R. (2011). 10 lessons from 10 years of measuring and modeling the internet's autonomous systems. *IEEE Journal on Selected Areas in Communications*, 29(9) :1810–1821.

[Schölkopf et al., 2000] Schölkopf, B., Herbrich, R., and Smola, A. (2000). Relationships between gaussian processes, support vector machines and smoothing splines. *Technical Report, Institute for Adaptive and Neural Computation, University of Edinburgh*.

[Schölkopf et al., 2001] Schölkopf, B., Herbrich, R., and Smola, A. (2001). A generalized representer theorem. *Lecture Notes in Computer Science*, 2111 :416–426.

[Schölkopf and Smola, 2002] Schölkopf, B. and Smola, A. (2002). *Learning with Kernels*. MIT Press.

[Schoukens et al., 2003] Schoukens, J., Nemeth, J. G., Crama, P., Rolain, Y., and Pintelon, R. (2003). Fast approximate identification of nonlinear systems. *Automatica*, 39(7) :1267–1274.

[Sebakor et al., 2009] Sebakor, M., Theera-Umpon, N., and Auephanwiriyakul, S. (2009). Centralized control system in interdomain routing environments. In *Intelligent Signal Processing and Communications Systems, 2008. ISPACS 2008. International Symposium on*, pages 1–4.

[Sjöberg et al., 1993] Sjöberg, J., McKelvey, T., and Ljung, L. (1993). On the use of regularization in system identification. In *Proceedings of the 12th IFAC World Congress*, volume 7, pages 318–386, Sydney, Australia.

[Sjöberg et al., 1995] Sjöberg, J., Zhang, Q., Ljung, L., Benveniste, A., Delyon, B., Glorennec, P., Hjalmarsson, H., and Juditsky, A. (1995). Nonlinear black-box modeling in system identification : A unified overview. *Automatica*, 31(12) :1691–1724.

[Söderström and Stoica, 1988] Söderström, T. and Stoica, P. (1988). *System identification*. Prentice-Hall, Inc.

[Solak et al., 2003] Solak, E., Murray-Smith, R., Leithead, W. E., Leith, D. J., and Rasmussen, C. E. (2003). Derivative observations in gaussian process models of dynamic systems. In *Advances in neural information processing systems*, pages 1057–1064.

[Souza, 2010] Souza, C. (2010). Kernel functions for machine learning applications. Technical report.

[Spinelli et al., 2005] Spinelli, W., Piroddi, L., and Lovera, M. (2005). On the role of prefiltering in nonlinear system identification. *IEEE Transactions on Automatic Control*, 50 :1597–1602.

[Steinwart and Christmann, 2008] Steinwart, I. and Christmann, A. (2008). *Support vector machines*. Springer Science & Business Media.

[Steinwart et al., 2006] Steinwart, I., Hush, D., and Scovel, C. (2006). An explicit description of the reproducing kernel hilbert spaces of gaussian RBF kernels. *IEEE Transactions on Information Theory*, 52 :4635–4643.

[Stock and Watson, 2003] Stock, J. H. and Watson, M. W. (2003). *Introduction to econometrics*, volume 104. Addison Wesley Boston.

[Suykens et al., 2002] Suykens, J., Gestel, T. V., Brabanter, J. D., Moor, B. D., and Vandewalle, J. (2002). *Least Squares Support Vector Machines*. Singapore : World Scientific.

[Suykens et al., 2001] Suykens, J. A., Vandewalle, J., and De Moor, B. (2001). Optimal control by least squares support vector machines. *Neural networks*, 14(1) :23–35.

[Tarantola, 2005] Tarantola, A. (2005). *Inverse Problem Theory and Methods for Model Parameter Estimation*. SIAM.

[Tenorio, 2001] Tenorio, L. (2001). Statistical regularization of inverse problems. *SIAM review*, 43(2) :347–366.

[Tiels and Schoukens, 2013] Tiels, K. and Schoukens, J. (2013). From coupled to decoupled polynomial representations in parallel wiener-hammerstein models. In *Decision and Control (CDC), 2013 IEEE 52nd Annual Conference on*, pages 4937–4942. IEEE.

[Tikhonov and Arsenin, 1977] Tikhonov, A. and Arsenin, V. (1977). *Solutions of Ill-Posed Problems*. Winston/Wiley.

[Tóth, 2010] Tóth, R. (2010). *Modeling and identification of linear parameter-varying systems*. Lecture Notes in Control and Information Sciences, Vol. 403, Springer, Heidelberg.

[Tóth et al., 2011] Tóth, R., Laurain, V., Zheng, W. X., and Poolla, K. (2011). Model structure learning : A support vector machine approach for LPV linear-regression models. In *Proceedings of the 50th IEEE Conference on Decision and Control*, pages 2561 – 2566, Orlando, Florida, USA.

[Tych et al., 2002] Tych, W., Pedregal, D., Young, P., and Davies, J. (2002). An unobserved component model for multi-rate forecasting of telephone call demand : the design of a forecasting support system. *International Journal of Forecasting*, 18 :673–695.

[Valenzuela et al., 2015] Valenzuela, P. E., Rojas, C. R., and Hjalmarsson, H. (2015). A graph theoretical approach to input design for identification of nonlinear dynamical models. *Automatica*, 51 :233–242.

[Van Den Hof and Schrama, 1995] Van Den Hof, P. M. and Schrama, R. J. (1995). Identification and control—closed-loop issues. *Automatica*, 31(12) :1751–1770.

[Van Overschee and De Moor, 2012] Van Overschee, P. and De Moor, B. (2012). *Subspace identification for linear systems : Theory—Implementation—Applications*. Springer Science & Business Media.

[Vapnik, 1998] Vapnik, V. (1998). *Statistical Learning Theory*. Wiley.

[Wahba, 1990] Wahba, G. (1990). Spline models for observational data. In *Proceedings of the SIAM CBMS-NSF Regional Conference Series in Applied Mathematics*, volume 59, Philadelphia, Pennsylvania, USA.

[Weickert, 1998] Weickert, J. (1998). *Anisotropic Diffusion in Image Processing*. B.G. Teubner Stuttgart.

[Wernholt and Gunnarsson, 2006] Wernholt, E. and Gunnarsson, S. (2006). Nonlinear identification of a physically parameterized robot model 1. *IFAC Proceedings Volumes*, 39(1) :143–148.

[Wooldridge, 2015] Wooldridge, J. M. (2015). *Introductory econometrics : A modern approach*. Nelson Education.

[Yang and Rong, 2015] Yang, D. and Rong, Z. (2015). Evolution of the internet at the autonomous system level. In *34th Chinese Control Conference (CCC)*, pages 1313–1317, Hangzhou, China.

[Young, 2011] Young, P. (2011). *Recursive Estimation and Time-Series Analysis : An Introduction for the Student and Practicioner*. Springer-Verlag.

[Young et al., 1999] Young, P., Pedregal, D., and Tych, W. (1999). Dynamic harmonic regression. *Journal of Forecasting*, 18 :369–394.

[Young and Taylor, 2012] Young, P. and Taylor, C. J. (2012). Recent developments in the captain toolbox for matlab. In *16th IFAC Symposium on System Identification*, Bruxelles, Belgium.

[Zhou, 2002] Zhou, D. (2002). The covering number in learning theory. *Journal of Complexity*, 18 :739–767.

[Zhou, 2008] Zhou, D. (2008). Derivative reproducing properties for kernel methods in learning theory. *Journal of Computational and Applied Mathematics*, 220(1-2) :456–463.

[Zhou et al., 1995] Zhou, K., Doyle, J., and Glover, K. (1995). *Robust and Optimal Control*. Prentice Hall.