A Comparison of Parameter Estimation Methods for Linear Systems on Finite Intervals

Boshra Badran

Department of Electrical & Computer Engineering

McGill University

Montreal, Canada

April 2021

2021/04/13

A thesis submitted to McGill University in partial fulfillment of the requirements for the degree of Master of Science.

Abstract

Dynamical systems are found in almost every field, and Ordinary differential equations (ODEs) are often used to model these systems. By solving the ODE, we can predict the behaviour of the system. However, some ODEs are not solvable via analytical and numerical approaches. For such cases, practitioners leverage approaches that estimate the parameters of the ODE. Recently, many methods have been developed which estimate the ODE parameters, and these methods vary in their estimation accuracy, computation time, and noise handling. Moreover, due to the variety of available parameter estimation methods, practitioners are faced with the challenge of selecting the best method for their needs.

Therefore, in this thesis, we compare the performance of four cutting-edge parameter estimation methods, namely the **Parameter Cascades method**, **Joint Estimation method**, **Two-step method**, and finally, **IV-GLS method**. To evaluate the methods, we leverage a linear ODE system with varying noise levels to estimate its parameters using each of the methods. Moreover, we compute three performance metrics, namely the computation time, L2 norm and infinity norm. The evaluation results show that the IV-GLS method provides the most accurate estimation of the parameters, even with high levels of noise. On the other hand, the Two-step method requires the shortest computation time to provide its estimation. Our results provide guidance to practitioners when deciding which parameter estimation method to use for their dynamical system.

Abrégé

Les systèmes dynamiques se trouvent dans presque tous les domaines, et les équations différentielles ordinaires (EDO) sont souvent utilisées pour modéliser ces systèmes. En résolvant l'EDO, on peut prédire le comportement du système. Cependant, certains EDO ne peuvent pas être résolus par des approches analytiques et numériques. Pour tels cas, les praticiens utilisent des approches qui estiment les paramètres de l'EDO. Récemment, beaucoup de méthodes ont été développées pour estimer les paramètres EDO, et ces méthodes se varient en termes de précision de l'estimation, de temps de calcul et de gestion du bruit. De plus, à cause de la variété des méthodes d'estimation des paramètres disponibles, les praticiens sont confrontés au défi de choisir la meilleure méthode pour leurs besoins.

Donc, dans cette thèse, on compare les performances de quatre méthodes d'estimation de paramètres en pointe, c'est-à-dire la méthode **Les Cascades de paramètre**, la méthode **d'estimation conjointe**, la méthode **Deux-Etape** et finalement la méthode **VI-MCG** (Variables instrumentales - moindres carrés généralisée). Pour évaluer les méthodes, on utilise un système EDO linéaire avec différents niveaux de bruit pour estimer ses paramètres en utilisant chacune des méthodes. De plus, on calcule trois métriques de performance, à savoir le temps de calcul, la norme L2 et la norme à l'infini. Les résultats de l'évaluation montrent que la méthode VI-MCG donne l'estimation la plus précise des paramètres, même avec des niveaux de bruit élevés. Par contre, la méthode Deux-Etape nécessite le temps de calcul le plus court pour donner son estimation. Nos résultats aident les praticiens à décider de quelle méthode d'estimation des paramètres à utiliser pour leur système dynamique.

Acknowledgment

First and foremost, I would like to thank God for the countless blessings bestowed upon me and the incredible people I have in my life. To my supervisor Prof. Hannah Michalska who provided me with the guidance I needed to go through this journey, and inspired me to prove myself in my field, you have my absolute gratitude and appreciation.

To my loving and supportive parents, Abdullah and Rabaa, whom I received their continuous support in the past couple of years, I will be forever grateful for your existence in my life. Thank you for giving me the strength to reach for the stars and chase my dreams. To my beloved brother Khaled, who supported me in my ups and downs, thank you for always being there when I needed you. I cannot forget the long days we spent together studying and working hard in the library. To my sister Rama, who always stood by me in the rough times and celebrated with me the good ones. I know that your future will be bright, and I wish you all the best in your life. To my youngest siblings, Amer and Ranim, thank you for offering me your invaluable support and love.

I would also like to thank my friend Walaa, with whom I shared the undergraduate journey. We promised each other to continue and learn as much as we can. Thank you for encouraging me even when we are continents apart. Our journeys will stay entangled long after this. To my friends in Montreal, Hind and Luna, I am lucky to have friends like you. I have a special place in my heart for the memories we made together. And finally, to the rest of my family and friends, I am sincerely thankful for all of you.

Preface

This is to declare that the research carried in this thesis was completed by Boshra Badran under the supervision of Professor Hannah Michalska. To the best of our knowledge, this is the first work to compare the performance of these cutting-edge parameter estimation methods for linear Ordinary Differential Equations on finite time intervals.
The four methods are: the Parameter Cascades method developed by Cao and Ramsay [6] and Ramsay et al. [31], the Joint Estimation method developed by Carey [7], the Two-step method developed by John [22], and finally the IV-GLS method that was developed by Ghoshal and Michalska [15].

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Ordinary Differential equations (ODEs) have a central role in a multitude of applications as they can describe complex dynamical systems. ODEs can model systems from various domains, ranging from medicine to mechanical and electrical engineering and more. Some ODE's can be solved using analytical or numerical methods. However, a significant portion of ODEs is not solvable using these approaches. In such cases, practitioners often leverage a parameter estimation method that predicts the system behaviour over time. In other words, parameter estimation methods enable practitioners to obtain an understanding of a diverse set of dynamical systems that are unsolvable via conventional approaches. This explains the broad interest of the research community in these methods. In fact, many methods have been developed by researchers in recent years to estimate a system's parameters. For example, the Artificial Neural Network approximation-based decomposition method that was developed by Dua [10] and applied to systems that model various chemical reactions. Another example is the multiple shooting method [26], which was applied to estimate the parameters of oscillations in calcium signalling.

However, practitioners are faced with a challenge when selecting the best method to estimate their system's parameters due to the variety of available methods. For example, to predict the behaviour of a dynamical system for HIV, Liang and Wu [23] used the

pseudo–least squares method, while Holder and Rodrigo [19] used the integration based method. In both cases, practitioners attempted to estimate the parameters of the same system, but ended up selecting different parameter estimation methods.

Hence, this thesis aims to provide an in-depth comparison of four cutting-edge parameter estimation methods to evaluate their performance, especially with regards to the methods' ability to handle systems with varying noise levels. The methods we asses are: **Parameter Cascades method**, developed by Ramsay et al. [31] and Cao and Ramsay [6]; **Joint Estimation method**, put forward by Carey et al. [8]; **Two-step method**, introduced by John [22]; and finally, **IV-GLS method**, which was developed by Ghoshal and Michalska [15].

To evaluate these methods, we leverage a single input single output linear time-invariant system. More specifically, we use the methods to estimate this system's parameters and compare their results with the ground truth parameters and trajectories. Moreover, to assess the method's ability to handle noise in the system, we test the methods using the same system, but with varying amounts of Additive White Gaussian Noise (ranging from $\sigma = 0$ to 1.5 with $\mu = 0$). We also measure several widely-used performance metrics for parameter estimation methods, namely, the computation time, L2 norm, and infinity norm. Furthermore, We study the effects of increasing the noise levels and the sample points in the system on the methods' performance.

The thesis provides the following contributions:

- To the best of our knowledge, this is the first work to evaluate these cutting-edge methods using the various performance metrics mentioned above.
- We assess the methods using varying amounts of noise in the system.
- We explore the impact of increasing the number of sample points on the method's performance.

The rest of the thesis is organized as follows: Chapter 2 provides a brief description of the evolution of parameter and state estimation of dynamical systems in history. Chapter 3 explains in detail the mathematical approaches and the theory behind four widely-used parameter methods that will be considered in this thesis. Chapter 4 outlines the experiment setup and presents the mathematical system and metrics that are used in the evaluation process. Chapter 5 is devoted to the results of the evaluation. Chapter 6 explicates the results from the previous chapter, and presents the recommended parameter estimation method. Chapter 7 concludes the thesis and suggests ways for improvements in the future.

Chapter 2

Literature Review

Dynamical systems evolve with time, and typically, to model these dynamical systems, Ordinary Differential Equations (ODEs) are used. Oftentimes, we are interested in predicting the behaviour of the target system for an extended time period. To obtain these predictions, we need to estimate the parameters of the ODE accurately. However, due to the expensive computational effort required to solve the ODE numerically and the unavailability of analytical solutions for most ODEs, estimating the ODE's parameters presents a significant challenge. Nevertheless, many methods have been developed over the years to solve this estimation problem. These methods used different approaches, and they varied in their success and performance.

One of the earliest methods that attempted to estimate a system's parameters is the nonlinear least square (NLS) method introduced by Bard [3] in 1974. Biegler et al. [4] then used this method in 1986 to solve a real-life system from an industrial environment. The NLS method estimates the ODE parameters by minimizing the differences between the observed data and the estimated data with respect to the estimated parameter. However, this method employs an iterative procedure to reach its goal, and for that, this method needs to have suitable initial values; otherwise, the iteration will either not converge, or it will converge to a local (vs. a global) minimum.

To overcome the minimization problem of the NLS, Gelman et al. [13] introduced the Bayesian approach. Recently Putter et al. [28], Huang and Wu [20], and Huang et al. [21] further improved the Bayesian approach by introducing hierarchical Bayesian approaches. Nevertheless, [28, 20, 21] applied the improved method on HIV dynamic models to estimate their parameters.

Another method to estimate a model's parameters was the two-stage method first suggested by Varah [36] in 1982. This method was later developed into the iterated principal differential analysis method (iPDA) by Heckman and Ramsay [18], Ramsay and Silverman [30], and Poyton et al. [27]. The iPDA method uses the spline technique to fit the discrete measurements of the output variables. First, the dynamical equation's derivative trajectories are estimated and substituted as variables into the ODEs; then, nonlinear least square is used to estimate the ODE parameters in the second stage. The drawback of this method is its complexity and the difficulty of estimating the derivatives.

Motivated by the previous method, Ramsay et al. [31], and Cao and Ramsay [6] proposed the generalized profiling method, also called parameter cascade and profiling method. This method uses a multi-criterion optimization scheme and penalized spline smoothing technique to estimate the ODE parameters. The estimate of the parameters is usually accurate; however, this method is also computationally expensive.

To overcome the computational performance drawback of profiling with the parameter cascading method, Carey et al. [8] proposed a Generalized smoother for linear ODEs. In this method, the partial derivatives are solved explicitly with respect to the parameters. Then, an iterative strategy is employed to get the optimal estimates of the parameters.

Fliess et al. [12] and Sira-Ramírez et al. [34] depended on a different approach for estimating the parameters by leveraging numerical differentiation, where differentiators are used instead of nonlinear asymptotic estimators. Inspired by the differentiation method, Ghoshal et al. [16] developed the kernel representations for differential systems method. Later on, Ghoshal and Michalska [14] and Sridhar et al. [35] continued working on the Kernel

method where they developed it into a method that can identify and reconstruct a dynamic system's trajectory from a single realization of an output measurement on a finite interval. Finally, Ghoshal and Michalska [15] extended the kernel method by using instrumental variable and covariance weighting to enable the method to estimate the parameters and trajectories from the noisy measurements accurately.

Chapter 3

Details of the Evaluated Parameter Estimation Methods

We initiate our study by selecting the parameter estimation methods that will be evaluated and compared in this thesis. All methods will be applied on linear systems, and they function on finite time intervals. In the following, we list the parameters estimation methods that fit our criteria and which will be evaluated in this study:

- **Parameter Cascades method**: A method developed by James Ramsay and Jiguo Cao to estimate liner systems' parameters. In essence, this method uses a multi-criterion optimization scheme and penalized spline smoothing technique to estimate three levels of parameters. The local parameters define the coefficient of the basis functions that defined the smooth, the global parameters which define the roughness penalty, and lastly, the complexity parameters that control the amount of roughness in the smooth.

- **Joint Estimation method**: Put forward by Michelle Carey, Eugene Gath, and Kevin Hayes, this method solves the estimation problem by forming a model-based penalty, deriving the mathematical term for local, global, and complexity parameters,

and finally using an iterative strategy to estimate the optimal parameters.

- **Two-step method**: A method introduced by Anju John. This method uses double sided kernel and features a simple unconstrained minimization to identify and estimate the system's actual parameter from single noisy realizations on a finite time interval.

- **IV-GLS method**: Debarshi Ghoshal and Hannah Michalska developed this parameter estimation method. This method depends also on the kernel representation of the differential invariance combined with a feasible recursive version of the generalized least squares, covariance weighting, and instrumental variables to accurately estimate the parameters from noisy measurements.

In the next sections, we further explain the details of each method and the steps we followed to evaluate the methods.

3.1 Parameter Cascades Method

This method was developed by Ramsay et al. [31] and Cao and Ramsay [6] to estimate functional parameters. The method transforms most functional data and models to matrix expressions by representing the functions in the form of linear combination of basis functions expansions. A basis function system is a set of known functions ϕ_k that are mathematically independent of each other, and that have the property that we can well approximate any function by taking a weighted sum or linear combination of a sufficiently large number K of these functions.

$$x(t) = \sum_{k=1}^{K} c_k \phi_k(t) \tag{3.1}$$

The expansion can capture any amount of variation in the data. Using so many basis functions will lead to over fitting; to avoid over fitting, roughness penalties of the below

form are used. Roughness penalties are based on optimizing a fitting criterion that defines what a smoothing of the data is trying to achieve.

$$P(x) = \lambda \int [Lx(t)]^2 \mathrm{d}t \tag{3.2}$$

where L is a linear differential operator of order m of the form:

$$Lx(t) = \sum_{j=0}^{m-1} \beta_j(t) D^j x(t) + D^m x(t) \tag{3.3}$$

So there is a need to minimize

$$\begin{aligned} J(c|\beta, \lambda) &= \sum_{j}^{n} [y_j - x(t_j)]^2 + \lambda \int [Lx(t)]^2 \mathrm{d}t \\ &= \sum_{j}^{n} [y_j - \mathbf{c}'\phi(t_j)]^2 + \mathbf{c}'\mathbf{R}(\lambda)\mathbf{c} \end{aligned} \tag{3.4}$$

where

$$\mathbf{R}(\lambda) = \lambda \int \mathbf{L}\phi(\mathbf{t})\mathbf{L}\phi(\mathbf{t})' \mathrm{d}t \tag{3.5}$$

A good and thoughtful selection of the roughness penalties will lead to better estimates of the parameters.

From the above equations, we have three parameters that need to be estimated:

- c_k which can be called a nuisance, incidental or local parameters. They are the coefficients of the basis function expansion defining the smooth. "The coefficients c_k are data-proxies in the sense that, if a saturated B-spline basis is used, then each coefficient represents the typical size of the observations y_i over the bounded interval over which the kth basis function is non-negative. The number of c_k is not fixed, and it can vary according to the characteristics of the data".

- β, which are structural or global parameters that need to be constants, as changing their values will impact all of the fit $x(t)$. In this thesis, these parameters of the

linear system are the ones that we estimate using the different methods to compare the methods' performance.

- λ which is the complexity parameter that controls the amount of roughness in the smooth.

After identifying the parameters that need to be estimated, and by following the previous steps, the multi criterion optimization problem becomes as follows:

1. Estimate $\hat{c}$ by minimizing the inner criterion $J(c|\beta, \lambda)$ in (3.4) with respect to c. It is typically based on any suitable measure of the quality of the fit to the data in addition to a regularization or smoothing term that defines smoothness in terms of c. Moreover, the $\hat{c}$ estimate is guided by the complexity parameter λ and possibly also on one or more of the global parameters in β.

   vector $\hat{c}$ will now become:

$$\hat{\mathbf{c}}(\beta, \lambda) = [\mathbf{\Phi}'\mathbf{\Phi} + \mathbf{R}(\beta, \lambda)]^{-1}\mathbf{\Phi}'\mathbf{y} \tag{3.6}$$

   where

$$\mathbf{R}(\beta, \lambda)(\lambda) = \lambda \int \mathbf{L}\phi(\mathbf{t})\mathbf{L}\phi(\mathbf{t})'\mathrm{d}t \tag{3.7}$$

   Ramsay et al. [31] and Cao and Ramsay [6] utilized a quadrature rule, namely, the trapezoidal rule to approximate the integrals in $\mathbf{R}(\beta, \lambda)$, thus the second term in (3.4) is also an error sum of squares. Hence, (3.4) is re-expressed as a non-linear least-squares problem, and the Gauss-Newton algorithm is employed to perform this optimisation.

   This step defines the local parameters as vector-valued functions of the global parameter β and the complexity parameter λ, causing the local parameters c to lose their status as independent parameters. This process of defining a subset of the parameters as functions of other parameters is called *profiling*,

2. Estimate β by minimizing the middle criterion $H(c(\beta)|\beta(\lambda))$ that defines the fit to the data conditional on λ

$$\mathbf{H}(\beta|\lambda) = \sum_{j}^{n}[y_j - \hat{x}(t_j, \beta|\lambda)]^2 = \sum_{j}^{n}[y_j - \hat{\mathbf{c}}(\beta, \lambda)'\phi(t_j)]^2$$
$$= \mathbf{y}'[I - \mathbf{A}(\beta|\lambda)]'[I - \mathbf{A}(\beta|\lambda)]\mathbf{y} \tag{3.8}$$

where the n × n smoothing matrix $\mathbf{A}(\beta|\lambda) = \mathbf{\Phi}[\mathbf{\Phi}'\mathbf{\Phi} + \mathbf{R}(\beta, \lambda)]^{-1}\mathbf{\Phi}'$.

$H(\beta|\lambda)$ is un-penalized criterion as the roughness penalty term is dropped in it. This cancellation of the penalty term happens because the fitting function is already smoothed when minimizing $J(c|\beta, \lambda)$ and so it does not need to be regularized twice. By minimizing the un-penalized criterion $H(\beta|\lambda)$, the estimate of $\hat{\beta}$ is obtained for any value of λ, hence $\hat{\beta}$ is an implicit function of λ. Lastly, the structural parameter vector β is removed from the parameter space through the middle optimization as a function of λ.

3. Select the complexity parameter λ by minimizing the outer criterion $F(\hat{\beta}(\lambda), \lambda)$ that is based on the generalized cross validated criterion which is

$$F(\hat{\beta}(\lambda), \lambda) = GCV(\hat{\beta}(\lambda), \lambda) = n\frac{SSE(\hat{\beta}(\lambda), \lambda)}{[\text{dfe}(\hat{\beta}(\lambda), \lambda)]^2} \tag{3.9}$$

where degrees of freedom measure dfe(λ) is

$$\text{dfe}(\lambda) = n - trace[\mathbf{A}(\lambda)]$$

By this definition, F should be considered as a function of λ since $\hat{\beta}$ is an implicit function of λ. This definition of F also utilizes the standard GCV, which is an error sum of squares discounted for the degrees of freedom invested in the model. The

standard GCV permits us to determine the λ value by optimizing a fit measure that is relatively insensitive to complexity.

The final parameter estimates become the functional cascade $\hat{\mathbf{c}}[\hat{\beta}(\hat{\lambda})], \hat{\beta}(\hat{\lambda}), \hat{\lambda}$ defined by the optimizations with respect to criteria J, H and F, respectively. Notably, The distinct hierarchy in these three classes of parameters is defined as *parameter cascade*.
As described in this method, the unique aspect of the parameter cascade is that a different criterion is optimized for each of the three levels of the cascade. That is, three-criterion optimization problem need to be solved $\mathbf{J}(c|\beta, \lambda)$, $\mathbf{H}(\beta|\lambda)$, and $\mathbf{F}(\lambda)$ for the local, global and complexity parameters, respectively. At the base level of the computations lie a large number of local parameters in c, and these are defined here as functions $\hat{\mathbf{c}}(\beta, \lambda)$ of the global and complexity parameters through the estimate (3.6). Even if an explicit solution for $\hat{\mathbf{c}}$ were not possible, this function can still be defined implicitly by optimizing a penalized or regularized fitting function like $\mathbf{J}(c|\beta, \lambda)$ each time the value of a global parameter change. When the functional relationship is implicit in this way, and the regularity conditions that are required to ensure that the optimization problem has a unique solution hold, the Implicit Function Theorem permits the explicit gradient and Hessian calculations which are essential for fast optimization.

For the third step, selecting the complexity parameter, the optimization of $F(\hat{\beta}(\lambda), \lambda)$ becomes faster and more stable if we have the gradient

$$\frac{\mathrm{d}F(\hat{\beta}(\lambda), \lambda)}{\mathrm{d}\lambda} = \frac{\partial F(\hat{\beta}(\lambda), \lambda)}{\partial \lambda} + \frac{\partial F(\hat{\beta}(\lambda), \lambda)}{\partial \hat{\beta}} \frac{\partial \hat{\beta}}{\partial \lambda} \tag{3.10}$$

where $\mathrm{d}F(\hat{\beta}(\lambda), \lambda)/\mathrm{d}\lambda$ is the total derivative of F with respect to λ. "We notice that the formula of $\mathrm{d}F(\hat{\beta}(\lambda), \lambda)/\mathrm{d}\lambda$ involves the term $\partial\hat{\beta}/\partial\lambda$. If the middle optimization leads to an explicit solution for $\hat{\beta}(\lambda)$, the gradient is readily available. But if not, the Implicit Function Theorem can be applied to find $\partial\hat{\beta}/\partial\lambda$. Since the optimal local parameter vector

$\hat{\beta}$ satisfies $\partial H(\beta|\lambda)/\partial\beta = 0$, and $\hat{\beta}$ is a function of λ and y, we can take the λ-derivative on $\partial H(\beta|\lambda)/\partial\beta \mid_{\hat{\beta}} = 0$ as follows:

$$\frac{\mathrm{d}}{\mathrm{d}\lambda}\left(\frac{\partial H(\beta|\lambda)}{\partial\beta} \mid_{\hat{\beta}}\right) = \frac{\partial^2 H(\beta|\lambda)}{\partial\beta\partial\lambda} \mid_{\hat{\beta}} + \frac{\partial^2 H(\beta|\lambda)}{\partial\beta^2} \mid_{\hat{\beta}} \frac{\partial\hat{\beta}}{\partial\lambda} = 0 \tag{3.11}$$

which holds since $\partial H(\beta|\lambda)/\partial\beta \mid_{\hat{\beta}}$ is a function of λ that is identically 0". Now assuming that $\partial^2 H(\beta|\lambda)/\partial\beta^2 \mid_{\hat{\beta}} \neq 0$, from the Implicit Function Theorem we get:

$$\frac{\partial\hat{\beta}}{\partial\lambda} = -\left[\frac{\partial^2 H(\beta|\lambda)}{\partial\beta^2} \mid_{\hat{\beta}}\right]^{-1}\left[\frac{\partial^2 H(\beta|\lambda)}{\partial\beta\partial\lambda} \mid_{\hat{\beta}}\right] \tag{3.12}$$

3.2 Joint Estimation Method

This method was developed by Michelle Carey and presented in her thesis Carey [7] and in the research paper Carey et al. [8]. The steps of this method are described below.

3.2.1 General steps

Linear homogeneous ODEs are defined by associating $f(x)$ with its rates of change

$$D^p f(x) = \sum_{r=0}^{p-1} -\beta_r D^r f(x) \tag{3.13}$$

where $D^r f(x)$ is the r^{th} derivative of $f(x)$ and β_r is the parameter that describes the relationship between the r^{th} and p^{th} rate of change. $f(x)$ is approximated by a linear combination of basis functions $f(x) = B(x)c$, where the $n \times K$ matrix $B(x)$ contains the underlying basis functions evaluated at the locations x, and the vector c contains the K corresponding coefficients [29] [30]. For modeling an applicable system, the ODE is selected such that its solution is the theorized form of $f(x)$ and is incorporated into the estimation procedure in the form of a model-based penalty R(β), which is given by the L2 norm of the ODE [27] [18].

The goal of this method is to estimate three different parameters: c which are the coefficients of the basis function expansion $B(x)c$; β which are the p parameters of the ODE, and λ, the complexity parameter, which controls the fitted curves $f(x)$ adherence to the solution of the ODE. Moreover, in order to achieve its goal, this method has to minimize the following objective function:

$$PENSSE(y|x, c, \beta, \lambda) = \|y - B(x)c)\|^2 + \lambda c' R(\beta) c \tag{3.14}$$

Now, to solve for (3.13) and (3.14), this method forms the model based penalty R(β), derives the mathematical term for $c(\beta, \lambda)$, $\beta(c)$, $\lambda(c, \beta, \rho)$, and finally uses the iterative strategy to estimate the optimal $\hat{c}$, $\hat{\beta}$, and $\hat{\lambda}$. Next, to get an updating method for $\hat{c}$, $\hat{\beta}$, and $\hat{\lambda}$, the partial derivative of $PENSSE(y|x, c, \beta, \lambda)$ with respect to c, the partial derivative of $PENSSE(y|x, c, \beta, \lambda)$ with respect to β, and the partial derivative of $GCV(\beta, \lambda)$ with respect to λ will be solved. Then, we use the iterative strategy that starts with small values of λ with the updating method to get the optimal values for $\hat{c}$, $\hat{\beta}$, and $\hat{\lambda}$.

3.2.2 Detailed steps

The linear ODE in (3.13) is incorporated into the penalized residual sum-of-squares in (3.14) in the form of a model-based penalty, which is given by the L2 norm of the ODE.

$$\begin{aligned} PENSSE(y|x, c, \beta, \lambda) &= \|y - B(x)c)\|^2 + \\ \lambda \int_{\pi_0}^{\pi_K} &\left[D^p f(x) + \sum_{r=0}^{p-1} \beta_r D^r f(x) \right]^2 dx \end{aligned} \tag{3.15}$$

$f(x)$ is approximated by a linear combination of basis functions $f(x) \approx B(x)c$; the coefficients of the basis function expansion c are obtained by minimizing (3.15) with respect to c for a fixed β and λ; the p parameters of the ODE are obtained by minimizing (3.15) with respect to β for a fixed c, and the complexity parameter which controls the fitted curves

$f(x)$ adherence to the solution of the ODE is obtained by minimizing the GCV criterion given by

$$GCV(\beta, \lambda) = \frac{n \left\| y - B(x)(B(x)'B(x) + \lambda R(\beta))^{-1} B(x)'y \right\|^2}{\left[n - trace(B(x)(B(x)'B(x) + \lambda R(\beta))^{-1} B(x)') \right]^2} \tag{3.16}$$

with respect to λ for fixed c and β. where $\mathrm{R}(\beta) = \int_{\pi_0}^{\pi_K} \left[D^p f(x) + \sum_{r=0}^{p-1} \beta_r D^r f(x) \right]^2 dx$

3.2.2.1 Construction of the Model-Based Penalty

"Linear L-spline basis functions are linear differential operators that are essentially linear combinations of the derivatives of order m B–spline basis functions." Given any linear homogeneous differential equation of the form

$$D^p f(x) = \sum_{r=0}^{p-1} -\beta_r D^r f(x)$$

one can rearrange the terms and, without loss of generality, set $\beta_p(x) = 1$ to produce a linear differential operator $Lf(x)$ that is represented by:

$$Lf(x) = \sum_{r=0}^{p} \beta_r(x) D^r f(x)$$

In here, the L2 norm of the linear differential operator can be calculated using:

$$\|Lf(x)\|^2 = \int_{\pi_0}^{\pi_K} \left[\sum_{r=0}^{p} \beta_r D^r f(x) \right] \left[\sum_{s=0}^{p} \beta_s D^s f(x) \right] dx \tag{3.17}$$

Next, by approximating the regression function by a B–spline basis function expansion $f(x) \approx \sum_{k=1}^{K} c_k B_k(x)$ on the interval [0, 1], we obtain the following equivalent approximation for the linear differential operator:

$$\|Lf(x)\|^2 = \sum_{k,j=1}^{K} c_k c_j \int_{\pi_0}^{\pi_K} \left[\sum_{r=0}^{m-1} \beta_r D^r B_{k,m} \right] \left[\sum_{s=0}^{m-1} \beta_s D^s B_{j,m} \right] dx \tag{3.18}$$

As an order m B–spline has only $m - 1$ non-zero derivatives, p should be at most $m - 1$. Thus the sums on r and s in equation (3.18) may be extended to $m - 1$. If

$p < m-1$, then we define $\beta_r \equiv 0$ for $p < r \leq m-1$. In practice, differential equations may not contain all the derivatives $D^r f(x)$ for $r = 0, ..., p-1$. Equation (3.18) can be easily modified for these cases by setting the coefficient $\beta_r(x) \equiv 0$ for the r^{th} derivative that is not contained in the differential equation. Next, substituting the basis function expansion for $f(x)$, $f(x) = \sum_{k=1}^{K} c_k B_{k,m}(x)$, into the L_2 norm of the linear ODE in (3.13) results in:

$$\begin{aligned}
&\int_{\pi_0}^{\pi_K} \left[D^p f(x) + \sum_{r=0}^{p-1} \beta_r D^r f(x) \right]^2 dx \\
&= \int_{\pi_0}^{\pi_K} \left[\sum_{k=1}^{K} c_k D^p B_{k,m}(x) + \sum_{r=0}^{p-1} \beta_r \sum_{k=1}^{K} c_k D^r B_{k,m}(x) \right] \\
&\quad \times \left[\sum_{l=1}^{K} c_l D^p B_{l,m}(x) + \sum_{s=0}^{p-1} \beta_s \sum_{l=1}^{K} c_l D^s B_{l,m}(x) \right] dx \\
&= c' \left[(I_K \otimes \beta') \Psi (I_K \otimes \beta) + 2\Omega' (I_K \otimes \beta) + \Phi \right] \times c \\
&= c' R(\beta) c
\end{aligned} \tag{3.19}$$

where $\beta = (\beta_0, ..., \beta_{p-1})'$, Ψ is a $(Kp) \times (Kp)$ matrix with $K \times K$ submatrices each of which is a $p \times p$ matrix

$$\psi_{k,l} = \begin{pmatrix} \int_{\pi_0}^{\pi_k} D^0 B_{k,m}(x) D^0 B_{l,m}(x) dx & \cdots & \int_{\pi_0}^{\pi_k} D^0 B_{k,m}(x) D^{p-1} B_{l,m}(x) dx \\ \vdots & \ddots & \vdots \\ \int_{\pi_0}^{\pi_k} D^{p-1} B_{k,m}(x) D^0 B_{l,m}(x) dx & \cdots & \int_{\pi_0}^{\pi_k} D^{p-1} B_{k,m}(x) D^{p-1} B_{l,m}(x) dx \end{pmatrix}$$

Ω' is a $K \times (Kp)$ matrix with $K \times K$ submatrices each of which is a $1 \times p$ vector

$$\begin{aligned}
\eta'_{k,l} = \frac{1}{2} \Bigg[&\left(\int_{\pi_0}^{\pi_k} D^p B_{l,m}(x) D^0 B_{k,m}(x) dx, \cdots, \int_{\pi_0}^{\pi_k} D^p B_{l,m}(x) D^{p-1} B_{k,m}(x) dx \right) + \\
&\left(\int_{\pi_0}^{\pi_k} D^p B_{k,m}(x) D^0 B_{l,m}(x) dx, \cdots, \int_{\pi_0}^{\pi_k} D^p B_{k,m}(x) D^{p-1} B_{l,m}(x) dx \right) \Bigg]
\end{aligned}$$

Φ is a $K \times K$ matrix containing the constants

$$\varphi_{k,l} = \int_{\pi_0}^{\pi_k} D^p B_{k,m}(x) D^p B_{l,m}(x) dx$$

Explicit expressions for each of the above terms is in [5].

3.2.2.2 An Expression for $c(\beta, \lambda)$

Now that we have constructed the model-based penalty, we set out to develop an expression for $c(\beta, \lambda)$. To that goal, we start by substituting the penalty matrix expression in (3.19) into (3.15), which gives:

$$\begin{aligned} PENSSE(y|x, c, \beta, \lambda) &= \|y - B(x)c)\|^2 + \lambda c' R(\beta) c \\ &= \|y - B(x)c\|^2 + \lambda c' [(I_K \otimes \beta')\Psi(I_K \otimes \beta) + 2\Omega'(I_K \otimes \beta) + \Phi] c \end{aligned} \tag{3.20}$$

Now by differentiating this penalised residual sum-of-squares with respect to c, for a given β, yields

$$\frac{\partial}{\partial c} PENSSE(y|x, c, \beta, \lambda) = 2B(x)'B(x)c - 2B(x)'y + 2\lambda R(\beta)c$$

Setting this expression equal to zero gives

$$(B(x)'B(x) + \lambda R(\beta)) = B(x)'y$$

If the matrix $(B(x)'B(x) + \lambda R)$ is invertible, then the estimate $\hat{c}$ of c is given by

$$\hat{c}(\beta, \lambda) = (B(x)'B(x) + \lambda R(\beta))^{-1} B(x)'y \tag{3.21}$$

3.2.2.3 An Expression for $\beta(c)$

The next parameter that we will develop an expression for in this method is $\beta(c)$. And similarly to $c(\beta, \lambda)$, we start by setting the partial derivative of the penalized residual

sum-of-squares in (3.20) with respect to β equal to zero and solving for β which results in:

$$\frac{\partial}{\partial \beta_j} PENSSE(y|x, c, \beta, \lambda) = \lambda c' \frac{\partial R(\beta)}{\partial \beta_j} c$$

$$= \lambda c' \frac{\partial}{\partial \beta_j} [(I_K \otimes \beta')\Psi(I_K \otimes \beta) + 2\Omega'(I_K \otimes \beta) + \Phi]c$$

$$= 2\lambda c'(I_K \otimes e_j{}')\Psi(I_K \otimes \beta)c + 2\lambda c'(I_K \otimes e_j{}')\Omega c \tag{3.22}$$

for $j = 0, ..., p-1$, where e_j is a vector of length p with the j^{th} entry equal to one, and all other entries equal to zero (i.e., $(e_j)_i = \delta_{i,j}$). Now, by setting (3.22) equal to zero we get:

$$c'(I_K \otimes e_j{}')\Psi(I_K \otimes \beta)c + c'(I_K \otimes e_j{}')\Omega c = 0 \tag{3.23}$$

In order to solve (3.23) for β, the equation can be expressed as

$$\sum_{i=0}^{p-1} [c'(I_K \otimes e_j{}')\Psi(I_K \otimes \beta_i e_i)c] = -c'(I_K \otimes e_j{}')\Omega c$$

And this can be written in vector form as:

$$(I_p \otimes c')A(I_p \otimes c)\beta = -(I_p \otimes c')\Gamma c$$

where A is the $(Kp) \times (Kp)$ matrix with the (i, j) entry

$$A_{i,j} = (I_K \otimes e_i{}')\Psi(I_K \otimes e_j)$$

and Γ is $(Kp) \times 1$ matrix with j^{th} entry

$$\Gamma_j = (I_K \otimes e_j{}')\Omega$$

In here, if the $p \times p$ matrix $(I_p \otimes c')A(I_p \otimes c)$ is invertible, then the estimate $\hat{\beta}$ of β is

$$\hat{\beta}(c) = -[(I_p \otimes c')A(I_p \otimes c)]^{-1}(I_p \otimes c')\Gamma c \tag{3.24}$$

Typically in this case, $(p+1)$ will be smaller than four.

3.2.2.4 An Expression for $\lambda(c, \beta, \rho)$

The final parameter that needs to be tackled is $\lambda(c, \beta, \rho)$, and to do so, we initially setting the partial derivative of the GCV criterion in (3.16) with respect to λ equal to zero and solving for λ resulting in:

$$\hat{\lambda}(c, \beta, \rho) = \frac{-c'R(\beta)c - \rho c'B(x)'B(x)[G(c)'G(c)]^{-1}G(c)'}{\rho c'R(\beta)[G(c)'G(c)]^{-1}G(c)'} \tag{3.25}$$

where $G(c) = (y - B(x)c)'B(x)$. Here, We introduce a constant parameter ρ which is a function of c, β, λ and is given by:

$$\rho(c, \beta, \lambda) = \frac{(y - B(x)c)^2 trace[-B(x)(B(x)'B(x) + \lambda R(\beta))^{-1}R(\beta)(B(x)'B(x) + \lambda R(\beta))^{-1}B(x)']}{n - trace[B(x)(B(x)'B(x) + \lambda R(\beta))^{-1}B(x)']} \tag{3.26}$$

For the GCV criterion we let $u = n \left\| y - B(x)c \right\|^2$ and $v = [n - trace(S(\beta, \lambda))]^2$. Now, differentiating equation (3.16) with respect to λ yields:

$$\frac{\partial GCV(\beta, \lambda)}{\partial \lambda} v \frac{\partial u}{\partial c}\frac{\partial c}{\partial \lambda} - u\frac{\partial v}{\partial \lambda} \tag{3.27}$$

where

$$\frac{\partial u}{\partial c} = -2n(y - B(x)c)'B(x)$$

$$\frac{\partial c}{\partial \lambda} = -(B(x)'B(x) + \lambda R(\beta))^{-1}R(\beta)c + (B(x)'B(x) + \lambda R(\beta))^{-1}$$

$$\frac{\partial v}{\partial \lambda} = -2(n - trace(S(\beta, \lambda)))trace\left(\frac{\partial S(\beta, \lambda)}{\partial \lambda}\right)$$

$$\frac{\partial S(\beta, \lambda)}{\partial \lambda} = -B(x)(B(x)'B(x) + \lambda R(\beta))^{-1}R(\beta)(B(x)'B(X) + \lambda R(\beta))^{-1}B(x)'$$

Next, setting equation (3.27) equal to zero produces:

$$\begin{aligned} nv(y - B(x)c)'B(x)[-(B(x)'B(x) + \lambda R(\beta))^{-1}R(\beta)c + (B(x)'B(x) + \lambda R(\beta))^{-1}] \\ -u(n - trace(S(\beta, \lambda)))trace(\frac{\partial S(\beta, \lambda)}{\partial \lambda}) = 0 \end{aligned} \tag{3.28}$$

let

$$\rho = \frac{u(n - trace(S(\beta, \lambda)))trace(\frac{\partial S(\beta, \lambda)}{\partial \lambda})}{nv}$$

and

$$G = (y - B(x)c)'B(x$$

By multiplying both sides of equation (3.28) by G' we get:

$$G'G[-(B(x)'B(x) + \lambda R(\beta))^{-1}R(\beta)c + (B(x)'B(x) + \lambda R(\beta))^{-1}] = \rho G'$$

Now if $G'G$ is invertible then,

$$-(B(x)'B(x) + \lambda R(\beta))^{-1}R(\beta)c + (B(x)'B(x) + \lambda R(\beta))^{-1}] = \rho(G'G)^{-1}G'$$

Multiplying both sides of the equation by $(B(x)'B(x) + \lambda R(\beta))^{-1}$ produces

$$-R(\beta)c = \rho(B(x)'B(x) + \lambda R(\beta))(G'G)^{-1}G'$$

Lastly, gathering the terms involving λ yields:

$$-R(\beta)c - \rho B(x)'B(x)(G'G)^{-1}G' = \lambda\rho R(\beta)(G'G)^{-1}G'$$

By multiply both sides of the equation by c' and rearranging the terms, we got the final estimate of λ represented as:

$$\hat{\lambda}(c,\beta,\rho) = \frac{c'[-R(\beta)c + v\beta + \varphi] - \rho c'B(x)'B(x)[G(c)'G(c)]^{-1}G(c)'}{\rho c'R(\beta)[G(c)'G(c)]^{-1}G(c)'} \tag{3.29}$$

3.2.2.5 An Iterative Scheme for Estimating $\hat{c}$, $\hat{\beta}$ and $\hat{\lambda}$

The steps of the iterative scheme were taken from Carey et al. [8]. We now have point estimates for $\hat{c}$, $\hat{\beta}$ and $\hat{\lambda}$. Here, we provide an iterative scheme that obtains optimal estimates of β the linear ODE parameters, c the coefficients of the B-spline basis function and λ the complexity parameter.

- **Step #1** Define initial estimates for all the parameters. Set $\hat{\lambda}^{(0)}$ so that neither the measures for the data adherence nor the fidelity differential equation dominate (3.20). By default we set $\hat{\lambda}^{(0)}$ =1.
  Get $\hat{c}^{(0)}$ by p$^{\text{th}}$ order smoothing and substitute $\hat{\lambda}^{(0)}$ for $\lambda\hat{c}^{(0)} = (B(x)'B(x)+\hat{\lambda}^{(0)}\phi)^{-1}B(x)'y$.
  Substitute $\hat{c}^{(0)}$ for c in (3.24) to get $\hat{\beta}^{(0)} = [(I_p \otimes \hat{c'}^{(0)})A(I_p \otimes \hat{c}^{(0)})]^{-1}(I_p \otimes \hat{c'}^{(0)})[-\Gamma\hat{c}^{(0)}]$.

- **Step #2** Attain the estimated complexity parameter. To do so we use:

$$\hat{\rho}^{(i)} = \frac{(y-B(x)\hat{c}^{(i)})^2 trace[-B(x)(B(x)'B(x)+\lambda^{(i)}R(\hat{\beta}^{(i)}))^{-1}R(\hat{\beta}^{(i)})(B(x)'B(x)+\lambda^{(i)}R(\hat{\beta}^{(i)}))^{-1}B(x)']}{n-trace[B(x)(B(x)'B(x)+\lambda^{(i)}R(\hat{\beta}^{(i)}))^{-1}B(x)']}$$

and

$$\hat{\lambda}^{(i+1)}(\hat{c}^{(i)},\hat{\beta}^{(i)},\hat{\rho}^{(i)}) = \frac{\hat{c'}^{(i)}[-R(\hat{\beta}^{(i)})\hat{c}^{(i)}] - \hat{\rho}^{(i)}\hat{c'}^{(i)}B(x)'B(x)[G'(\hat{c}^{(i)})G(\hat{c}^{(i)})]^{-1}G'(\hat{c}^{(i)})}{\hat{\rho}^{(i)}\hat{c'}^{(i)}R(\hat{\beta}^{(i)})[G'(\hat{c}^{(i)})G(\hat{c}^{(i)})]^{-1}G'(\hat{c}^{(i)})}$$

- **Step #3** Obtain the estimated coefficients of the basis function expansion and the estimated parameters of the ODE. The expressions for the parameters are:

$$\hat{c}^{(i+1)}(\hat{\beta}^{(i)}, \hat{\lambda}^{(i+1)}) = (B(x)'B(x) + \hat{\lambda}^{(i+1)}R(\hat{\beta}^{(i)}))^{-1} \times [B(x)'y]$$

and

$$\hat{\beta}^{(i+1)}(\hat{c}^{(i+1)}) = [(I_p \otimes \hat{c'}^{(i+1)})A(I_p \otimes \hat{c}^{(i+1)})]^{-1} \times (I_p \otimes \hat{c'}^{(i+1)})[-\Gamma\hat{c}^{(i+1)}]$$

- **Step #4** Repeat step 2 and step 3 until $GCV(\hat{\beta}^{(i)}, \hat{\lambda}^{(i+1)})$ reaches a minimum. As we achieve optimal convergence for c and β when λ is fixed. After GCV reaches a minimum, we set the estimated complexity parameters to the current value and repeat Step 3 until the parameters $\hat{c}^{(i+1)}(\hat{\beta}^{(i)}, \hat{\lambda}^{(i+1)})$ and $\hat{\beta}^{(i)}(\hat{c}^{(i)})$ converge.

Finally, since the penalized residual sum-of-squares in (3.20) is a positive definite quadratic form in both β and c for fixed $\hat{\lambda}^{(i-1)}$, it follows that:

$$PENSSE(y|x, \hat{c}^{(i)}, \hat{\beta}^{(i)}, \hat{\lambda}^{(i)}) \geq PENSSE(y|x, \hat{c}^{(i)}, \hat{\beta}^{(i+1)}, \hat{\lambda})$$

for all $\hat{\beta}$, and that:

$$PENSSE(y|x, \hat{c}^{(i)}, \hat{\beta}^{(i)}, \hat{\lambda}^{(i)}) \geq PENSSE(y|x, \hat{c}^{(i+1)}, \hat{\beta}^{(i)}, \hat{\lambda}^{(i)})$$

for all $\hat{c}$. Moreover, since this is a positive non-increasing sequence, it converges to a limit. Thus, the estimates converge and $(\hat{\beta}^{(i+1)}, \hat{c}^{(i+1)})$ as $i \to \infty$ are the minimum estimates of the least squares criterion in (3.20) with respect to β and c.

3.3 Two-step Method

John [22] developed this method by building on the work of Ghoshal et al. [17] that introduced a simpler approach for deriving the algebraic estimation equations. In essence, this method transforms a high order differential equation that includes a system invariant into an integral expression with no singularities at the boundaries of the time interval. This transformation is done by applying the Cauchy formula for multiple integrals and the forward and backward integration. To characterize system trajectories, this method leverages the reproduced output of the system invariance in any time interval. In the following, we explain how we apply this method to a single-input single-output (SISO) linear time invariant (LTI) system.

3.3.1 A Double Sided Kernel in SISO LTI Representation

To estimate a system's parameter using this method, we first need to reach an expression for the differential invariant of the system. To do so, we consider a SISO LTI system of the form:

$$\begin{aligned} \dot{x} &= Ax \\ y &= Cx \end{aligned} \tag{3.30}$$

the matrix A of the system is in canonical structure,

$$\begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ -a_0 & -a_1 & -a_2 & \dots & -a_{n-1} \end{bmatrix} \tag{3.31}$$

and,

$$C = \begin{bmatrix} 1 & 0 & 0 & \dots \end{bmatrix} \tag{3.32}$$

Thus, the system's characteristic equation is,

$$y^{(n)}(t) + a_{n-1}y^{(n-1)}(t) + \cdots + a_1 y^{(1)}(t) + a_0 y(t) = 0 \tag{3.33}$$

Now, we need to successfully identify the unknown values of the parameters $a_i, i = 0, .., n-1$ given noisy observations of the system's output $y(t)$ over an arbitrary finite interval of time $t \in [0, T]$, $T > 0$. For a homogeneous system, the estimation of the parameter is equivalent to identifying the differential invariant $\mathcal{I}$ ($\mathcal{I} \equiv 0$) of a dynamical system. It is important to note here that the differential invariant remains the same under the action of the system dynamics. The differential invariant $\mathcal{I}$ can be expressed as:

$$\begin{aligned}&\mathcal{I}(t, y(t), y^{(1)}(t), \cdots, y^{(n)}(t)) \\ &\quad = y^{(n)}(t) + a_{n-1}y^{(n-1)}(t) + \cdots + a_0 y(t)\ ;\ t \geq 0\end{aligned} \tag{3.34}$$

One issue that arises from the conventional integral of (3.34) is the singularity at $t = 0$. In this method, the problem of the singularity is avoided when a two-sided forward-backward integration is applied, resulting in an alternative integral system representation of (3.34) in a reproducing kernel Hilbert space (RKHS), as described in the following theorem.

Theorem 1 *There exist Hilbert-Schmidt kernels K_{DS}, K^i_{DS}, $i = 1, \cdots n-1$, such that the output function y of (3.34) is reproduced on any given interval $[a, b]$ in accordance with the action of the evaluation functional*

$$y(t) = \int_a^b K_{DS}(t, \tau) y(\tau)\ \mathrm{d}\tau\ ;\ \forall t \in [a, b] \tag{3.35}$$

and the derivatives of the output $y^{(1)}, \cdots y^{(n-1)}$ can be computed recursively by way of output integration, so that for $i = 1, \cdots n-1$ and for all $t \in [a, b]$:

$$y^{(i)}(t) = \sum_{k=0}^{i-1} b_k(t) y^{(k)}(t) + \int_a^b K^i_{DS}(t, \tau) y(\tau)\ \mathrm{d}\tau \tag{3.36}$$

where $y^{(0)} \equiv y$ *and* $b_k(\cdot)$ *are rational functions of* t. *Hilbert-Schmidt kernels are square integrable functions on* $L^2[a,b] \times L^2[a,b]$.

Knowing the system's characteristic equation allows us to use the reproducing property to get the output on an arbitrary finite interval of time $[a,b]$. The differential invariance is used to obtain the behavioural model of the system. Next, We can reconstruct the output and its time derivatives that validate the system invariance from noisy measurements using the mathematics of the Reproducing Kernel Hilbert Space (RKHS) of the behavioural model. Finally, we develop a fourth order kernal[1] for the system which we use in the next steps.

3.3.2 Parameter Estimation Using L-splines in RKHS

After developing the forward-backward kernal, we will define the minimization problem and solve it in a two steps approach (hence the method's name).

3.3.2.1 Optimization-based mathematical formulations for estimation problem[25]

Consider a general fourth-order system:

$$y^{(4)}(t) + a_3 y^{(3)}(t) + a_2 y^{(2)}(t) + a_1 y^{(1)}(t) + a_0 y(t) = 0 \tag{3.37}$$

The aim is to estimate the system's parameter vector $\vec{a}$ from noisy observation $z(\tau); \tau \in [a,b]$ of the function $y(\tau); \tau \in [a,b]$. The expression

$$L(y) = \int_a^b K_{DS}^a(t,\tau) y(\tau) d\tau \tag{3.38}$$

denotes an operator that maps the function $y \in L^2[a,b]$ into the RKHS where the kernel K_{DS} depends on the unknown parameter vector $\vec{a}$, with a positive symmetric kernel expressed as

$$K^a(t_1,t_2) = \int_a^b K_{DS}^a(t_1,\tau) K_{DS}^a(t_2,\tau) d\tau \tag{3.39}$$

[1]For a detailed explanation of the kernal development steps, please refer to [22].

The "noisy measurement function" $z(t)$,$t \in [a,b]$ is projected onto the subspace in RKHS defined by:

$$S^a \triangleq \{y \in Ker^{\perp}L \mid s.t\ y(t) = \int_a^b K^a_{DS}(t,z)y(z)dz\} \tag{3.40}$$

where, $Ker^{\perp}L$ denotes the orthogonal compliment to the null space of the operator L in (3.38). Next, we need to find the approximate expressions for projections $\hat{y} \in Ker^{\perp}L$ as finite linear combinations of the vectors in the subspace span$\{K_{DS}(t,.), t \in [a,b]\}$. These finite linear combinations will have the following form:

$$\hat{y}(.) \cong \sum_{j=1}^{n} c_j K^a_{DS}(t_j,\cdot) \tag{3.41}$$

for some distinct time instants $t_i \in [a,b], i = 1,...,n$ chosen in such a way that, $K^a_{DS}(t_j,\cdot)$ are linearly independent for $t_j, j = 1,...,n$ for distinct t_j. The linear combination in (3.41) should approximately satisfy the reproducing equation below:

$$\hat{y}(t) = \int_a^b K^a_{DS}(t,\tau)\hat{y}(\tau)d\tau \tag{3.42}$$

By substituting (3.41) in (3.42):

$$\begin{aligned}
\sum_{j=1}^{n} c_j K^a_{DS}(t_j,t) &= \int_a^b K^a_{DS}(t,\tau) \sum_{j=1}^{n} c_j K^a_{DS}(t_j,\tau)d\tau \\
&= \sum_{j=1}^{n} c_j \int_a^b K^a_{DS}(t,\tau)K^a_{DS}(t_j,\tau)d\tau \\
&= \sum_{j=1}^{n} c_j K^a(t,t_j) \\
&= \sum_{j=1}^{n} c_j K^a(t_j,t)
\end{aligned} \tag{3.43}$$

As RKHS kernels are symmetric, the above equation can be re-written as,

$$\sum_{j=1}^{n} c_j [K^a_{DS} - K^a](t_j,t) = 0 \tag{3.44}$$

or, equivalently,

$$c^T vec\{[K^a_{DS} - K^a](t_j, t)\}^n_{j=1} = 0 \tag{3.45}$$

with $c^T \triangleq [c_1, ..., c_n]$,

$$vec\{[K^a_{DS} - K^a](t_j, t)\}^n_{j=1} \triangleq [(K^a_{DS} - K^a)(t_1, t), ..., (K^a_{DS} - K^a)(t_n, t)]^T$$

The projection has to minimize

$$min\{\frac{1}{2}\|z - \hat{y}\|^2 \; ; c_j, j = 1, \cdots, n\} \tag{3.46}$$

subject to (3.44), where z represents the noisy observation. "Now, we need to compute the norm in (3.46) in the space $L^2[a, b]$. Recalling that the kernels K^a_{DS} and K^a both depend on the parameter vector a, to estimate the system parameter vector a, the arg min solution in (3.46) of the above must further (or simultaneously) be minimized w.r.t. a. For simplicity, the dependence of both K_{DS} and K on a is dropped from the notation in the sequel".

3.3.2.2 Step 1: Parameter estimation

we need to estimate the actual parameter a by making $V(a, z) = y$ using the minimization. If this equality is required at a finite number n of discrete-time points $t_j; j = 1, \cdots, n$, then the problem will be to obtain the optimal solution to:

$$\begin{aligned} &\min\{J(a) := \frac{1}{2n}\sum_{i=1}^{n}(y(t_i) - < y, K_{DS}(t_i, \cdot) >_2)^2 \mid \text{w.r.t.} \;\; a \in \mathbb{R}^3\} \\ &= \min\{\frac{1}{2n}\sum_{i=1}^{n}\left[y(t_i) - \int_a^b K_{DS}(t_i, \tau)y(\tau)d\tau\right]^2 \mid \text{w.r.t.} \;\; a \in \mathbb{R}^3\} \end{aligned} \tag{3.47}$$

$K_{DS}(t, \tau)$ is expressed as a scalar product of some partial kernels

$$K_{DS}(t, \tau) = K_v(t, \tau)^T a + k_{v5}(t, \tau) \;\; \text{equivalently} \;\; K_{DS}(t, \tau) = a^T K_v(t, \tau) + k_{v5}(t, \tau) \tag{3.48}$$

$$K_v(t,\tau)^T := [k_{v1}(t,\tau), k_{v2}(t,\tau), k_{v3}(t,\tau), k_{v5}(t,\tau)];\ a := [a_0, a_1, a_2, a_3]^T \tag{3.49}$$

By substituting the overhead expressions into the cost of (3.47) we get (with $T := b - a$)

$$\begin{aligned}
J(a) &:= \frac{1}{2n}\sum_{i=1}^{n}\left[y(t_i) - \int_a^b K_{DS}(t_i,\tau)\ y(\tau)d\tau\right]^2 \\
&= \frac{1}{n}\sum_{i=1}^{n}\left[\frac{1}{2}y(t_i)^2 - y(t_i)\int_a^b K_{DS}(t_i,\tau)\ y(\tau)d\tau + \frac{1}{2}\left(\int_a^b K_{DS}(t_i,\tau)\ y(\tau)d\tau\right)^2\right] \\
&= \frac{1}{n}\sum_{i=1}^{n}\left[\frac{1}{2}y(t_i)^2 - y(t_i)\int_a^b K_{DS}(t_i,\tau)\ y(\tau)d\tau + \frac{1}{2}\int_a^b K_{DS}(t_i,\tau)\ y(\tau)d\tau\int_a^b K_{DS}(t_i,s)\ y(s)ds\right] \\
&= \frac{1}{2n}\sum_{i=1}^{n}y(t_i)^2 - \frac{1}{n}\sum_{i=1}^{n}y(t_i)\int_a^b K_{DS}(t_i,\tau)\ y(\tau)\ d\tau \\
&\qquad + \frac{1}{2n}\sum_{i=1}^{n}\int_a^b\int_a^b K_{DS}(t_i,\tau)K_{DS}(t_i,s)\ y(\tau)y(s)\ d\tau\ ds
\end{aligned}$$

Expanding the kernels gives

$$\begin{aligned}
&J(a) = \\
&= \frac{1}{2n}\sum_{i=1}^{n}y(t_i)^2 - \frac{1}{n}\sum_{i=1}^{n}\int_a^b [K_v(t_i,\tau)a + k_{v5}(t_i,\tau)]\ y(\tau)y(t_i)\ d\tau \\
&\quad + \frac{1}{2n}\sum_{i=1}^{n}\int_a^b\int_a^b [a^T K_v(t_i,\tau) + k_{v5}(t_i,\tau)][K_v(t_i,s)^T a + k_{v5}(t_i,s)]\ y(\tau)y(s)\ d\tau\ ds \\
&= \frac{1}{2n}\sum_{i=1}^{n}y(t_i)^2 dt - \frac{1}{n}\sum_{i=1}^{n}\int_a^b K_v(t_i,\tau)^T\ y(\tau)y(t_i)\ d\tau\ a - \frac{1}{n}\sum_{i=1}^{n}\int_a^b k_{v5}(t_i,\tau)\ y(\tau)y(t_i)\ d\tau \\
&+ \frac{1}{2n}a^T\sum_{i=1}^{n}\int_a^b\int_a^b K_v(t_i,\tau)K_v(t_i,s)^T y(\tau)y(s)\ d\tau\ ds\ a \\
&+ \frac{1}{2n}\sum_{i=1}^{n}\int_a^b\int_a^b k_{v5}(t_i,\tau)K_v(t_i,s)^T\ y(\tau)y(s)\ d\tau\ ds\ a \\
&+ a^T\frac{1}{2n}\sum_{i=1}^{n}\int_a^b\int_a^b K_v(t_i,\tau)k_{v5}(t_i,s)\ y(\tau)y(s)\ d\tau\ ds \\
&+ \frac{1}{2n}\sum_{i=1}^{n}\int_a^b\int_a^b k_{v5}(t_i,\tau)k_{v5}(t_i,s)]\ y(\tau)y(s)\ d\tau\ ds
\end{aligned}$$

Assembling the terms above gives a standard quadratic:

$$J(a) = d + b^T a + \frac{1}{2} a^T C a \quad \text{with}$$

$$d := \left\{ \frac{1}{2n} \sum_{i=1}^{n} y(t_i)^2 - \frac{1}{n} \sum_{i=1}^{n} \int_a^b k_{v5}(t_i, \tau)\ y(\tau) y(t_i)\ d\tau \right.$$
$$\left. + \frac{1}{2n} \sum_{i=1}^{n} \int_a^b \int_a^b k_{v5}(t_i, \tau) k_{v5}(t_i, s)]\ y(\tau) y(s)\ d\tau\ ds \right\}$$

$$b^T := \left\{ -\frac{1}{n} \sum_{i=1}^{n} \int_a^b K_v(t_i, \tau)^T\ y(\tau) y(t_i)\ d\tau \right.$$
$$\left. + \frac{1}{2n} \sum_{i=1}^{n} \int_a^b \int_a^b [K_v(t_i, s)^T k_{v5}(t_i, \tau) + K_v(t_i, \tau)^T k_{v5}(t_i, s)]\ y(\tau) y(s)\ d\tau\ ds \right\}$$

$$C := \left\{ \frac{1}{n} \sum_{i=1}^{n} \int_a^b \int_a^b K_v(t_i, \tau) K_v(t_i, s)^T\ y(\tau) y(s)\ d\tau\ ds \right\}$$

The standard quadratic cost results in a minimization problem with respect to the parameter a. The problem is solved globally and analytically as follows:

$$\begin{aligned} &\min\{J(a) \mid a \in \mathbb{R}^3\} \quad \text{is attained globally and uniquely at} \\ &\hat{a} = -C^{-1} b; \quad \text{with minimum value} \quad J(\hat{a}) = d - \frac{1}{2} b^T C^{-1} b \end{aligned} \tag{3.50}$$

For simplicity, the double integrals above can be rewritten in terms of single integrals

$$\int_a^b \int_a^b k_{v5}(t_i, \tau) k_{v5}(t_i, s)]\ y(\tau) y(s)\ d\tau\ ds = \left(\int_a^b k_{v5}(t_i, \tau)\ y(\tau)\ d\tau \right)^2$$
$$\int_a^b \int_a^b [K_v(t_i, s)^T k_{v5}(t_i, \tau) + K_v(t_i, \tau)^T k_{v5}(t_i, s)]\ y(\tau) y(s)\ d\tau\ ds$$
$$= \left(\int_a^b K_v(t_i, s)^T y(s)\ ds \right) \left(\int_a^b k_{v5}(t_i, \tau)\ y(\tau)\ d\tau \right)$$

$$+\left(\int_a^b K_v(t_i,\tau)^T y(\tau)\ d\tau\right)\left(\int_a^b k_{v5}(t_i,s)\ y(s)\ ds\right)$$

$$\int_a^b\int_a^b K_v(t_i,\tau)K_v(t_i,s)^T y(\tau)y(s)\ d\tau\ ds$$

$$=\left[\int_a^b K_v(t_i,\tau)\ y(\tau)\ d\tau\right]\left[\int_a^b K_v(t_i,s)\ y(s)\ ds\right]^T$$

3.3.2.3 Step 2: Output and derivatives estimation by projection

In the second step of the approach, we reconstruct the system output trajectory by using:

$$\hat{y}=\int_a^b K_{DS}(t_i,\tau)z(\tau)d\tau \tag{3.51}$$

where $z(\tau)$ is the measured signal.

After estimating the parameter vector a, and having the solution of the characteristic equation, we can reconstruct the system output trajectory from a noisy measurement by direct orthogonal projection onto the subspace spanned by the fundamental solutions of the characteristic equation (3.33) of the system. Following that, we obtain the fundamental solutions by integrating (3.33), as implemented in the next steps.

First, we select initial conditions for the homogeneous LTI system in (3.33) in the form of n independent vectors which can be taken as the vectors of the canonical basis in $\mathbb{R}^n$:

$$e_1=[1,0,...,0]$$

$$e_2=[0,1,...,0]$$

$$\cdots \tag{3.52}$$

$$e_n=[0,0,...,1] \tag{3.53}$$

Then, we solve the system equation (3.33) for every particular initial condition, which produces:

$$\overline{y_i}(a)=e_i \qquad i=1,\cdots,n.$$

By this step, we have obtained the initial conditions for the ODE system in (3.33). We know that any solution of the ODE with any initial condition is a linear combination of such fundamental solutions $\overline{y_i}$. Hence, we have to obtain the coefficients of this linear combination that makes the resulting function the closest to the output measurement data. Now, we can find the closest solution with respect to the orthogonal projection onto the span of S^a.

$$S^a = \text{span}\left\{\overline{y}_i(\cdot), i = 1, \cdots, n\right\}$$

To do so, we need to orthonormaliz the set of fundamental solutions. The projection of a measured noisy signal $z(\cdot) \in L^2[a, b]$ into S^a is given as:

$$y_E(\cdot) \triangleq \arg\min \left\{ \|z - y\|_2^2 \, | y \in S^a \right\}. \tag{3.54}$$

Looking for:

$$\hat{y} = \sum_{i=1}^{n} \hat{c}_i \overline{y}_i \tag{3.55}$$

Since $\hat{y}$ is a linear combination, the optimality requirements in (3.54) are reached only if,

$$\left\langle z | \overline{y}_j \right\rangle_2 = \sum_{i=1}^{n} \hat{c}_i \left\langle \overline{y}_i | \overline{y}_j \right\rangle_2 j = 1, \cdots, n \tag{3.56}$$

where the matrix representation is:

$$\begin{aligned} &v = G(\overline{y})\hat{c}; \quad G(\overline{y}) \triangleq \text{mat}\left\{ \left\langle \overline{y}_i | \overline{y}_j \right\rangle_2 \right\}_{i,j=1}^{n} \\ &v \triangleq \text{vec}\left\{ \langle z | \overline{y}_i \rangle_2 \right\}_{i=1}^{n}; \hat{c} \triangleq \text{vec}\left\{ \hat{c}_i \right\}_{i=1}^{n} \end{aligned} \tag{3.57}$$

G is named the Gram matrix for vectors in span S^a. Since all fundamental solutions are linearly independent, the Gram matrix is invertible.

$$\hat{c} = G^{-1}(\overline{y})v \tag{3.58}$$

Therefore the aim of the second step, which is estimating the output trajectory $\hat{y}$, is attained from (3.55). Finally, after estimating $\hat{y}$, the derivatives of the reconstructed output trajectory can also be reconstructed using the equation:

$$y^{(i)}(t) = \int_a^b K^i_{DS}(t,\tau)\hat{y}(\tau)\mathrm{d}\tau \tag{3.59}$$

where, K^i_{DS} are the kernel representation for the derivatives of the output.

3.4 IV-GLS Method

This parameter estimation method was developed by Ghoshal and Michalska [15], and it leverages the system invariant in its implementation. The system's differential invariant is an expression that states the relationship between the system output and its derivatives and remains constant under the system flows. If the differential invariant of the system exists, we will be able to predict its output trajectory using this method.

Similar to the Two-step method, this method is based upon the idea of developing a kernel representation of the differential invariants. The distinctive double-sided kernel system representation had an obvious advantage of alleviating the singularity problem at the extremities of the estimation interval and dropping the need for re-initialization to compensate for the accumulation of numerical error in the fundamental estimation method. More importantly, the concept of kernel representation of differential invariants of LTI systems appeared with a bonus in the form of a reproducing kernel Hilbert space induced by the kernel themselves where the kernel functions could serve as a basis for projections and representations of bounded linear functions. While the kernel expressions showed effective parameter estimation under small measurement noise, the method presented here focuses on enhancing the kernel representation to compensate for higher measurement noise levels.

The finite interval estimation problem will have two main steps: first, estimate the parameters; second, reconstruct the output and the derivative trajectory from a single re-

alization of a measurement process with a large level of additive Gaussian noise. However, the resulting regression model fails to satisfy the Gauss-Markov Theorem's assumptions, calling for the use of approaches that can secure consistency and eliminate regression dilution due to error-in-the variable and heteroskedasticity. In this method, the latter problems are addressed by employing standard remedies with well established statistical and probabilistic properties such as the introduction of instrumental variables and a recursive version of the feasible generalized least squares with covariance weighting. In the following, we describe this method's steps and details.

3.4.1 A finite interval estimation problem for an LTI system

Suppose we have a Single Input Single Output Linear Time Invariant system of n-th orders, in a state space form developed on a given finite time interval $[a, b] \subset \mathbb{R}$:

$$\dot{x} = Ax + bu \ ; \quad y = c^T x \ ; \ x \in \mathbb{R}^n \tag{3.60}$$

the system matrices and the characteristic equation below have matching dimensions

$$\lambda^n + a_{n-1}\lambda^{n-1} + \cdots + a_1\lambda + a_0 = 0 \tag{3.61}$$

The input-output equation for system (3.60) becomes

$$y^{(n)}(t) + a_{n-1}y^{(n-1)}(t) + \cdots + a_1 y^{(1)}(t) + a_0 y(t) = \ - b_{n-1}u^{(n-1)}(t) - \cdots - b_0 u(t) \tag{3.62}$$

where $-b_i, i = 0, \ldots, n-1$ are the coefficients of the polynomial in the numerator of the rational transfer function for (3.60).

In order for this method to produce an estimate of a system's parameters, the following assumptions must hold:

(i) The input function $u(t)$, the derivatives of the input function $u^{(i)}(t), i = 1, \cdots, n-1$ for $t \in [a, b]$, and the dimension of the state vector of the Linear Time-Invariant system are known in advance.

(ii) The output of the system is recognized as a single realization of a 'continuous' measurement process $y_M(t) := y(t) + \eta(t)$, $t \in [a, b]$ in which η represents the additive white Gaussian noise (AWGN) with unknown variance σ^2. Having an unlimited number of output measurements over the finite interval $[a, b]$ is needed to satisfy this assumption.

Following these assumptions, the method is expected to:

(a) For the given time interval $[a, b]$, determine the identifiability of the input and output parameters of the system $a_i, b_i, i = 0, \cdots, n-1$ from a single realization of the measurement process $y_M(t), t \in [a, b]$;

(b) If the parameters are identified, the method needs to provide accurate estimates for the parameters $a_i, b_i, i = 0, \cdots, n-1$;

(c) For the given time interval $[a, b]$, estimate and reconstruct the true system output trajectory $y(t)$ and its $n-1$ time derivatives $y^{(i)}(t), i = 1, \ldots, n-1$ from the noisy observation;

(d) As there is an increment in the number of measurement samples of y_M, the error in the estimated output function y and its derivatives $y^{(i)}, i = 1, \ldots, n-1$ converges to zero consistently.

3.4.2 A kernel representation of a system differential invariance

This method depends on the integral representation of the controlled differential invariance of the system (3.62) which, for a homogeneous system, can be presented as a mapping J:

$$\begin{aligned} J(y, y^{(1)}, \cdots, y^{(n)}) &:= y^{(n)}(t) + a_{n-1}y^{(n-1)}(t) + \cdots + a_1 y^{(1)}(t) + a_0 y(t) \\ &\equiv const. = 0; \quad t \in [a, b] \end{aligned} \tag{3.63}$$

This mapping remains fixed when there is a flow of the system, and the external function does not exist. From the validity of (3.63), we can obtain more measurement-noise

independent information about the system's behaviour beyond the observation of the noisy y_M. Moreover, the characterization (3.63) of the homogeneous system has to have a configuration that does not rely on the system's initial conditions and does not require any time derivatives of the output, as the direct measurement cannot provide time derivatives.

The representation for the SISO LTI system depends on developing an integral kernel, as presented in the Theorems below.

Theorem 2 *There exist Hilbert-Schmidt kernels $K_{DS,y}$, $K_{DS,u}$, such that the input and output functions u and y of* (3.60) *satisfy*

$$y(t) = \alpha_{ab}^{-1}(t)\left[\int_a^b K_{DS,y}(n,t,\tau)y(\tau)d\tau + \int_a^b K_{DS,u}(n,t,\tau)u(\tau)d\tau\right] \tag{3.64}$$

where the inverse of the annihilator of the boundary conditions of the system has been chosen as

$$\alpha_{ab}^{-1}(t) := \frac{1}{(t-a)^n + (b-t)^n} \tag{3.65}$$

Hilbert-Schmidt double-sided kernels of (3.64) are square integrable functions on $L^2[a,b] \times L^2[a,b]$. The 'forward' and 'backward' kernels for the respective integral representations are stated below:

$$\begin{aligned} K_{DS,y}(n,t,\tau) &\triangleq \begin{cases} K_{F,y}(n,t,\tau), & \text{for } \tau \le t \\ K_{B,y}(n,t,\tau), & \text{for } \tau > t \end{cases} \\ K_{DS,u}(n,t,\tau) &\triangleq \begin{cases} K_{F,u}(n,t,\tau), & \text{for } \tau \le t \\ K_{B,u}(n,t,\tau), & \text{for } \tau > t \end{cases} \end{aligned} \tag{3.66}$$

The kernel functions $K_{DS,y}$, $K_{DS,u}$ have the property of being linear in the system parameters $a_i, b_i, i = 0, \cdots, n-1$ $n-1$, in addition, they are times differentiable as functions of t.

Remark 1. It is important to note that for $u \equiv 0$, the invariance representation (3.64) is in fact a continuous evaluation functional for the system output functions and hence induces a unique reproducing kernel Hilbert space (RKHS) with kernel

$$K_y(t_1, t_2) := \alpha_{ab}^{-2}(t) \int_a^b K_{DS,y}(t_1, \tau) K_{DS,y}(t_2, \tau) d\tau \tag{3.67}$$

$t_1, t_2 \in [a, b]$ where the dependence of the kernels on the system order n has been suppressed for brevity. Resulting from the remark, kernel $L(K_y(t, \cdot))$ is a representer for any enclosed linear functional $L : y \to L(y)(t)$ on the RKHS space. This indicates that for homogenous systems, for example, the kernel representation of the time derivatives of the output functions involves only the derivatives of the $K_{DS}(., \tau)$,

$$y^{(i)}(t) = \int_a^b \frac{\partial}{\partial t}[\alpha_{ab}^{-1}(t) K_{DS,y}(t, \tau)] y(\tau) d\tau \quad t \in (a, b) \tag{3.68}$$

The zero input solutions of the SISO LTI system are defined unambiguously as a novel characteristic of the reproducing property of the kernel demonstration of the system invariance. The reproducing feature (3.64) is mutual among all the fundamental solutions of the LTI system since they span a subspace of the RKHS.

From here, the method will perform the estimation problem in two steps: (1) Estimate the coefficients of the characteristic equation $a_i, i = 0, \ldots, n-1$ (parametric); (2) Reconstruct the system output trajectory and its time derivatives over the bounded time horizon $[a, b]$ (non-parametric).

3.4.3 Parametric estimation as a least squares problem

We can rewrite the reproducing property to highlight that the kernels of Theorem 2 are linear in the unknown system coefficients:

$$\begin{aligned} y(t) &= \int_a^b K_{DS,y}(t, \tau) y(\tau)\ d\tau \\ &= \sum_{i=0}^{n} \beta_i \int_a^b K_{DS(i),y}(t, \tau) y(\tau) d\tau \end{aligned} \tag{3.69}$$

where the $K_{DS(i),y}$; $i = 0, \cdots, n$ are 'component kernels' of $K_{DS,y}$ that post-multiply the coefficients $\beta_i = a_i$; $i = 0, \cdots, n-1$, with $\beta_n = 1$ for the support of the representation. In the noiseless case, the output variable y gets to be the evaluated output corresponding to the actual output trajectory y_T. As a consequence, the regression equation for the constant coefficients $a_i, i = 0, \ldots, n-1$, (3.69), can be written in a partitioned structure as:

$$y_T(t) = [K^{\bar{a}}, K^1](t; y_T)\beta^T \tag{3.70}$$
$$\bar{a} := [a_0, \cdots, a_{n-1}]; \quad \beta^T := [\bar{a}; \beta_n]$$

where $K^{\bar{a}}(t; y_T)$ is a row vector with integral components

$$K^{\bar{a}}(t; y_T)_k := \int_a^b K_{DS(k),y}(t, \tau) y_T(\tau) d\tau \; ; \; k = 0, \ldots n-1 \tag{3.71}$$

while $K^1(t; y_T)$ is a scalar

$$K^1(t; y_T) := \int_a^b K_{DS(n),y}(t, \tau) y_T(\tau) d\tau \tag{3.72}$$

corresponding to $\beta_n = 1$. If we have discrete time instants $t_1, \cdots, t_N \in (a, b]$, defined as *knots*, the regression equation is expressed point-wise in an equation that involves matrices

$$Q(y_T) = P(y_T)\bar{a} \tag{3.73}$$
$$Q := \begin{bmatrix} q(t_1) \\ \vdots \\ q(t_N) \end{bmatrix} \quad ; P := \begin{bmatrix} p_0(t_1) \cdots p_{n-1}(t_1) \\ \ddots \\ p_0(t_N) \cdots p_{n-1}(t_N) \end{bmatrix}$$
$$q(t_i) = y_T(t_i) - K^1(t_i, y_T);$$
$$p_k(t_i) = K^{\bar{a}}(t_i; y_T)_k \tag{3.74}$$

In the noiseless case, and if the identifiability hypothesis is satisfied, we can solve these equations using the least-squares error minimization.

3.4.3.1 Identifiability of homogeneous LTI systems from a single realization of a measured output

A homogeneous LTI system such as

$$\begin{aligned} &\dot{x}(t) = Ax(t); \quad y = Cx; \quad x \in \mathbb{R}^n \\ &x(0) = b \end{aligned} \tag{3.75}$$

is identifiable from a single noiseless recognition of its output trajectory y, given specific requirements, which are complicated to prove using computational methods. Nevertheless, the practical form of identifiability is satisfied with the given estimation scheme as described underneath.

Definition 1 (Practical linear identifiability) *The homogeneous system* (3.62) *is practically linearly identifiable on* $[a, b]$ *with respect to a particular noisy discrete realization of the output measurement process,* $y_M(t), t \in [a, b]$*, if and only if there exist distinct knots* $t_1, \cdots, t_N \in (a, b]$ *which render* $rank P(y_M) = n$*. Any such output realization is then called persistent.*

Moving to the cases where there is noise (consider it Additive White Gaussian Noise), the regression equation (3.69) should be replaced to have a stochastic regression problem, since the reproducing property fails to exist with the noisy output, which makes the equation no longer applicable. The reason is that the stochastic output determination works on a proper space, for example, $L^2(\Omega, \mathcal{F}, \mathbb{P})$ as the y_M that is supposed to accommodate the general filtration of the standard Wiener process W on $[a, b]$ is

$$y_M(t, \omega) = y_T(t) + \sigma \dot{W}(t, \omega) \; ; \quad t \in [a, b] \tag{3.76}$$

where y_T is the true output of the system and $\dot{W}$ implies the generalized derivative of the standard (one-dimensional) Wiener process; (see [32]). In other words, $\sigma \dot{W}$ is recognized with the white noise process with steady variance σ^2.

The kernel formulation:

$$\int_a^b K_{DS,y}(t,\tau)y_M(\tau)\ d\tau = \sum_{i=0}^{n}\beta_i \int_a^b K_{DS(i),y}(t,\tau)y_M(\tau)d\tau$$
$$= \int_a^b K_{DS,y}(t,\tau)y_T(\tau)\ d\tau + \int_a^b K_{DS,y}(t,\tau)\ \sigma\dot{W}(\tau)d\tau \tag{3.77}$$

is a random variable, and the subsequent equality exists because y_T fulfills the reproducing feature in the deterministic regression equation (3.69).

$$y_M(t) = \int_a^b K_{DS,y}(t,\tau)y_M(\tau)\ d\tau + e(t) \tag{3.78}$$

$$\text{with}\ \ e(t) := \sigma\dot{W}(t) - \int_a^b K_{DS,y}(t,\tau)\ \sigma\dot{W}(\tau)d\tau \tag{3.79}$$

The random error variable e is conditional on the unknown system parameters $a_i, i = 0,\cdots,n-1$, which is one thing that must be taken into consideration. The stochastic regression equation.

$$y_M(t) = \sum_{i=0}^{n}\beta_i \int_a^b K_{DS(i),y}(t,\tau)y_M(\tau)d\tau + e(t) \tag{3.80}$$

has the stochastic regressor vector

$$\left[\int_a^b K_{DS(0),y}(t,\tau)y_M(\tau)d\tau,\cdots,\int_a^b K_{DS(n),y}(t,\tau)y_M(\tau)d\tau\right]^T \tag{3.81}$$

From the above, and by checking the hypotheses of the Gauss-Markov Theorem, we can confirm that it is not applicable in the linear regression problem (3.80), as first the random regressor is associated with a regression error, and second, it fails to be homoskedastic.

Therefore the superior regression is a standard 'error-in-the-variable' dilemma with heteroskedastic noise.

3.4.3.2 Heteroskedasticity

The Ordinary least squares estimator has severe penalties when there is heteroskedasticity, such as wrong estimated regression error and inaccurate confidence intervals. On the other

hand, the OLS estimator remains unbiased. This method suggests an efficient approach to handle the *unknown* heteroskedasticity which is to employ GLS (generalized least squares), as the GLS estimator is proved to be BLUE (Best Linear Unbiased Estimator)[2].

The GLS applies inverse covariance weighting in the regression error minimization problem related to (3.80). As what was employed in the deterministic OLS (3.73), consider $Q(y_M)$ and $P(y_M)$ are the matrices corresponding to N instances of the measurement process realization y_M at a batch of knots $t_1, \cdots, t_N$. The yielded stochastic regression error vector is

$$e := [e(t_1), \cdots, e(t_N)]^T = Q(y_M) - P(y_M)\bar{a} \tag{3.82}$$

where $e(t_i)$ are as in (3.79). To estimate the parameter vector $\bar{a}$, the conventional GLS regression error minimization is

$$\begin{aligned} &\min_{\bar{a}} \left([Q(y_M) - P(y_M)\bar{a}]^T S [Q(y_M) - P(y_M)\bar{a}] \right) \\ &\text{with} \quad S := [\mathrm{Cov}(e)]^{-1}; \quad \mathrm{Cov}(e) := \mathrm{E}\left[ee^T\right] \end{aligned} \tag{3.83}$$

By applying the features of white noise and using the expectation operator to equations (3.76) and (3.79), the covariance computation carried out as below [15]:

$$\begin{aligned} \mathrm{Cov}[e(t_i), e(t_j)] = \sigma^2 \delta(t_i - t_j) - \sigma^2 K_{DS}(t_i, t_j) \sigma^2 K_{DS}(t_j, t_i) \\ + \sigma^2 \int_a^b K_{DS}(t_i, \tau) K_{DS}(t_j, \tau) d\tau \end{aligned}$$

Because the covariance matrix relies on the unknown variance σ^2 and the unknown parameter vector $\bar{a}$ in the K_{DS} kernels, the regular GLS as in (3.83) cannot be directly employed. To solve this issue, a *feasible* version of the GLS should be applied. Hence in this method, the covariance matrix is calculated gradually as more information about the regression residuals becomes obtainable. This is implemented as part of a recursive approach, where successive batches of samples are extracted from the realization of y_M.

Given that the regression error e_i is expressed as $Q_i - P_i\bar{a}$, the recursive GLS algorithm calculates:

$$\hat{a}_k = \arg\min_{\bar{a}} \left(\sum_{i=1}^{k} (Q_i - P_i\bar{a})^T S_i (Q_i - P_i\bar{a}) \right) \tag{3.84}$$

where $\hat{a}_k$ is the parameter estimate update at iteration k of the algorithm. After estimating the parameter $\hat{a}_k$ and the variance σ^2 that is gotten from the residual trajectory $y_M(t) - y_E(t)$ in preceding iteration k, the weighting matrix S_{k+1}, is calculated as the inverse of the covariance matrix.

3.4.3.3 Errors-in-variables

It is a known fact that the presence of errors-in-variables causes an asymptotic bias in OLS regression estimates, which is commensurable to the signal-to-noise ratio in the recognized regressand. The preeminent way to eliminate estimation bias in such cases is to use *Instrumental Variables* (IV) (see [38]) in the standard equations that produce the optimal estimates. The IV is formed using the backward reproducing kernels approach to provide statistical consistency for the estimation problem, which is described in what follows.

With regards to Theorem 2, the 'double-sided' regression equation (3.70) can be originated as two regression equations corresponding to the forward and backward kernels $K_{F,y}$ and $K_{B,y}$ as follows:

$$(t-a)^n y_T(t) = \int_a^t K_{F,y}(t,\tau)\ y_T(\tau) d\tau \tag{3.85}$$

$$:= [K_F^{\bar{a}}, K_F^1](t; y_T)\ \beta\ ; t \in [a,b] \tag{3.86}$$

$$(b-t)^n y_T(t) = \int_t^b K_{B,y}(t,\tau)\ y_T(\tau) d\tau \tag{3.87}$$

$$:= [K_B^{\bar{a}}, K_B^1](t; y_T)\ \beta\ ; t \in [a,b] \tag{3.88}$$

Having a set of knots $[t_1, \cdots, t_N]; N > n; t_1 >> a; t_N << b$, the forward and backward kernels $K_{F,y}$ and $K_{B,y}$ are written in discrete time using a matrix-vector form as N copies

of (3.86) and (3.88)

$$Y_T = K_F(y_T)\beta \tag{3.89}$$

$$Y_T = K_B(y_T)\beta \tag{3.90}$$

An OLS estimator for the parameter vector β corresponding to the normal equation is presented in (3.89) equation:

$$K_F(y_T)^T K_F(y_T)\beta = K_F(y_T)^T Y_T$$

namely:

$$\hat{\beta}_F := \left[K_F(y_T)^T K_F(y_T)\right]^{-1} K_F(y_T)^T Y_T \tag{3.91}$$

given that the inverted matrix is non-singular.

Next, we pre-multiply the forward estimation equation (3.89) by the backward matrix $K_B(y_T)$. If the matrix fulfills:

$$\det\left[K_B(y_T)^T K_F(y_T)\right] \neq 0 \tag{3.92}$$

then it delivers one more estimator which, in the noiseless case, will be competitive with that in (3.91):

$$K_B(y_T)^T K_F(y_T)\beta = K_B(y_T)^T Y_T$$

so that

$$\hat{\beta}_{IVF} := \left[K_B(y_T)^T K_F(y_T)\right]^{-1} K_B(y_T)^T Y_T \tag{3.93}$$

In the noiseless representation, both estimators provide exactly similar values for the estimated parameter vector, i.e. $\beta_F = \beta_{IVF}$, as long as the nonsingularity condition (3.92) is satisfied,

we can select a reasonable choice of the 'instrumental variable' using the fact that the forward and the backward kernel integrals (3.85)–(3.87) have just t as their point in

common, as they are defined on the intervals $[a,t]$ and $[t,b]$, respectively. K_B can be the IV for the forward equation (3.93), and K_F can be the IV for the backward equation. Hence, the forward equation (3.93) and the backward equation could be considered as 'instrumental variable estimators'.

The IV instrument's statistical features that were addressed earlier restore the consistency of the adjusted GLS estimator employing such IV instrument.

$$\hat{\beta}_{k+1} = \hat{\beta}_k + R_{k+1}\tilde{P}_{k+1}^T S_{k+1}(\tilde{Q}_{k+1} - \tilde{P}_{k+1}\hat{\beta}_k) \tag{3.94}$$

$$R_{k+1} = R_k - R_k\tilde{P}_{k+1}^T(S_{k+1}^{-1} + \tilde{P}_{k+1}R_k\tilde{P}_{k+1}^T)^{-1}\tilde{P}_{k+1}R_k \tag{3.95}$$

and the matrices P and Q are substituted by $\tilde{P}$ and $\tilde{Q}$ respectively, where

$$\tilde{P} = K_B(y_M)^T K_F(y_M)\ ;\ \tilde{Q} = K_B(y_M)^T Y_M \tag{3.96}$$

and $M_{k+1} = \sum_{i=0}^{k+1} \tilde{P}_i^T S_i \tilde{P}_i$ and $R_{k+1} = M_{k+1}^{-1}$.

Motivated by the fact that the last estimate for the recursive approach should lead to $\beta_n = 1$ where $\beta := [a_0, \cdots, a_{n-1}, \beta_n]$, a stopping condition is presented as

$$|\hat{\beta}_n - 1| < \epsilon \text{ for some } \epsilon > 0 \tag{3.97}$$

Finally, when applying this method, the error terms' covariance is calculated by substituting the K_{DS} by K_F or K_B properly.

3.4.4 Trajectory and derivative reconstruction

As the first step of solving the estimation problem is done, we now proceed to the subsequent step, estimating the output trajectory. We perform this step by projection onto the finite-dimensional subspace of the RKHS spanned by the fundamental solutions, expressed in this method by $\xi_1, \cdots, \xi_n$. More specifically, we follow these steps:

(a) Obtain the fundamental solutions by integrating the characteristic equation for n sets of initial conditions (where B_k are the canonical basis vectors in $\mathbb{R}^n$):

$$Y(0)_k := [y(0), y^{(1)}, \cdots, y^{(n-1)}] = B_k; \quad k = 1, \cdots, n \tag{3.98}$$

(b) Use the Gram-Schmidt ortho-normalization procedure in L^2, to orthonormalize the set $\xi_k, k = 1, \cdots, n$ into $\zeta_k, k = 1, \cdots, n$. As the ortho-normalizing transformation is linear with $\langle \cdot | \cdot \rangle_2$ as the inner products in L^2.

$$\begin{aligned} &\text{span}\{\xi_k, k = 1, \cdots, n\} = \text{span}\{\zeta_k, k = 1, \cdots, n\} := \Xi \\ &\langle \zeta_i | \zeta_j \rangle_2 = 0, i \neq j; \quad \langle \zeta_i | \zeta_i \rangle_2 = 1 \end{aligned} \tag{3.99}$$

(c) Estimate $\hat{y}_T$ with linear estimators $\hat{c}_i$ for the coefficients c_i in the form:

$$\hat{c}_i := \langle y_M | \zeta_i \rangle_2 = \int_a^b y_M(\tau)\zeta_i(\tau)d\tau, \quad i = 1, \cdots, n \tag{3.100}$$

as the true output trajectory is a linear combination of fundamental solutions

$$y_T = \sum_{i=1}^{n} c_i \zeta_i; \quad \text{with} \quad c_i = \langle y_T | \zeta_i \rangle_2, \quad i = 1, \cdots, n \tag{3.101}$$

(d) Reconstruct the output trajectory given $\bar{y}_M$ on the finite time interval $[a, b]$ using:

$$y_E(t) = \sum_{i=1}^{n} \langle \bar{y}_M | \zeta_i \rangle_2 \zeta_i(t); \quad t \in [a, b] \tag{3.102}$$

(e) Estimates the derivatives $y_E^{(i)}, i = 1, \cdots, n-1$ from y_E using these expressions:

$$\begin{aligned} y^{(i)}(t) &= \int_a^b K_{DS}^i(t, \tau) y_T(\tau) \mathrm{d}\tau \\ K_{DS}^i(t, \tau) &:= \left(\frac{\partial}{\partial t} \right)^i K_{DS}(t, \tau); \quad i = 1, \cdots, n-1 \end{aligned} \tag{3.103}$$

Finally, the estimator $\hat{y}_T$ that results from using the coefficients of (3.100) is BLUE as well. This arises from the fact that the estimator can be seen as an OLS where a set of measurements y_M is projected onto some subspace Ξ where the measurement errors are Gaussian i.i.d. Therefore, the strict Gauss-Markov assumptions are satisfied here.

Chapter 4

Experiment setup

In this chapter, we outline our experiment setup, showcase the system for the evaluation, and detail the metrics and approach used to recommend an optimal parameter estimation method.

Moving forward, it is essential to note that we exclude the Parameter Cascades method from our evaluation as Carey et al. [8] has shown that the Joint Estimation method produces comparable accuracy while at the same time being significantly less computationally exhausting. Moreover, Carey [7] proved that the sum square of errors of the predictions is less in the Joint Estimation method compared to the Parameter Cascades method.

4.1 Evaluation System

To evaluate the methods, we leverage a fourth order unstable system, which is described as:

$$\dot{x} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -1 & -5 & -5 & 0 \end{bmatrix} x \; ; y = x_1 \; ; \; x(0) = [0, 0, 0, 1] \tag{4.1}$$

and has the following characteristic equation:

$$y^{(4)}(t) + a_3 y^{(3)}(t) + a_2 y^{(2)}(t) + a_1 y^{(1)}(t) + a_0 y(t) = 0 \tag{4.2}$$

where the nominal (ground truth) values of the parameters are:

$$a_0 = 1\ , \quad a_1 = 5\ , \quad a_2 = 5\ , \quad a_3 = 0 \tag{4.3}$$

and the initial values are:

$$y(0) = 0\ , \quad y^{(1)}(0) = 0\ , \quad y^{(2)} = 0\ , \quad y^{(3)} = 1 \tag{4.4}$$

The system has two real roots, -0.63342 and -0.27912, and two complex ones, $0.45627 - 2.33404i$ and $0.45627 + 2.33404i$. Since there are poles in the right half-plane, the system is considered unstable. Our decision to use an unstable system was motivated by the fact that it is generally more challenging for parameter estimation methods to predict the parameters or control unstable systems [24].

We use each of the three methods to estimate this system's parameters and reconstruct its output trajectory within a finite interval from 0 to 6. To assess the methods' performance with noisy input, we carry out five different runs of the methods on the system, each with an increasing amount of noise. In the first run, the system is entirely noiseless, i.e., all the sample points represent the actual output trajectory. For the second, third, fourth, and fifth runs, we add to the true output of the system an Additive White Gaussian noise (AWGN) with a standard deviation $\sigma = 0.3$, $\sigma = 0.7$, $\sigma = 1$, $\sigma = 1.5$, respectively. Previous studies have used similar standard deviation σ values for the noise added to system trajectories. Moreover, by using a wide range of additive noise in our evaluation, we are able to represent real-life scenarios where systems can have anywhere from a small amount of noise to a significant amount of it.

Additive White Gaussian noise is characterized by the input output formula $y_k = x_k + n_k$ where the noise samples n_k are independent and identically distributed and drawn from a normal distribution with a zero mean $\mu = 0$ and a variance equal to $E = \{|N_k|^2\} = \sigma_n^2$.

The standard deviation that we are using in our evaluation is equal to the square root of the variance. [37] [11].

Finally, to run the experiment, we run a Matlab script to run each of the three evaluated methods against the system and analyze the methods' results to obtain the evaluation metrics.

4.2 Tracked Metrics

For each run and method, we record three important pieces of information: 1) **the computation time**, which is the time that it takes for the method to fully execute and produce its results. A shorter computational time means that the method is faster in estimating the result. 2) **L2 norm**, a measure of the distance between the vectors of the nominal and the reconstructed trajectory and it is calculated using the equation in (4.5), 3) **Infinity norm**, a measure of the maximum absolute differences between the vectors of the nominal and the reconstructed trajectory and it is calculated using the equation in (4.6).

$$\textbf{L2 Norm} = \left(\sum_{i=1}^{N} |y_T(t_i) - \hat{y}(t_i)|^2 \right)^{\frac{1}{2}} \tag{4.5}$$

$$\textbf{Infinity Norm} = Max\, |y_T(t_i) - \hat{y}(t_i)| \tag{4.6}$$

The L2 Norm and the Infinity norm metrics are special cases of the Minkowski metric (L_p), which is one of the most used quantitative distance measures in scientific and engineering applications [9]. We compute the Minkowski distance between the nominal trajectory vector and the reconstructed trajectory vector, specifically $\mathbf{y_T} = (y_T(t_1), y_T(t_2), ..., y_T(t_n))$ and $\hat{\mathbf{y}} = (\hat{y}(t_1), \hat{y}(t_2), ..., \hat{y}(t_n))$ in the n-dimensional Euclidean space, $\mathbb{R}^n$. the mathematical expression is given as:

$$L_p(\mathbf{y_T}, \hat{\mathbf{y}}) \left(\sum_{i=1}^{N} |y_T(t_i) - \hat{y}(t_i)|^p \right)^{\frac{1}{p}} \tag{4.7}$$

the Minkowski distance is the general formula, and some examples of it include the L_2 (Euclidean metric), and the L_∞ (chessboard metric) as in (4.5), and (4.6) respectively.

The L2 norm is more informative when assessing the states of the trajectories. On the other hand, the infinity norm is more informative when evaluating the accuracy of the parameters' estimation. The smaller the values of the L2 norm and the infinity norm, the more accurate the estimation of the trajectories' states and parameters is.

Chapter 5

Results

In this chapter we show the results of evaluating the Joint Estimation, Two-step, and IV-GLS methods.

5.1 Noiseless case

We consider the noiseless case to be our base case in this comparison. In this run, the sample points of the system exactly the same as those from the true output trajectory. Now, we apply each of the three methods on this noiseless system with the number of sample points $N = 50$, which were selected to be equidistant in $[0, 6]$. Table 5.1 presents the results of estimating the noiseless system's parameters from each method. The table also shows the elapsed time during each method's execution and the L2 norm and infinity norm metrics. The estimated parameters' values vary from one method to the other. However, all three methods result in estimated parameters with values that are quite close to the true parameters. The Joint Estimation method takes the minimum computation time compared to the Two-step and IV-GLS methods. However, both the Joint Estimation and IV-GLS produce more accurate estimations of the states of the trajectories and the parameters, with lower L2 and infinity norms compared to the Two-step method.

Figure 5.1 shows the true output trajectory with the three methods' reconstructed trajectories. The Joint Estimation and IV-GLS trajectories are almost identical to the actual trajectory, while there are small differences between the Two-step reconstructed trajectory and the true trajectory.

Method	**a0**	**a1**	**a2**	**a3**	**Computation time**	**L2 norm**	**Infinity norm**
True parameters	1	5	5	0			
Joint Estimation	0.9984	5.0076	4.9990	0.0005	33.60 s	1.6677e-04	7.9402e-05
Two-step	1.0084	4.7184	5.0222	-0.0508	33.776 s	0.1915	0.0790
IV-GLS	0.9999	4.9997	5.0002	0.0001	6 mins 32 s	0.0021	7.1406e-05

Table 5.1 Table showing true and approx. par values from a true output without noise and N=50

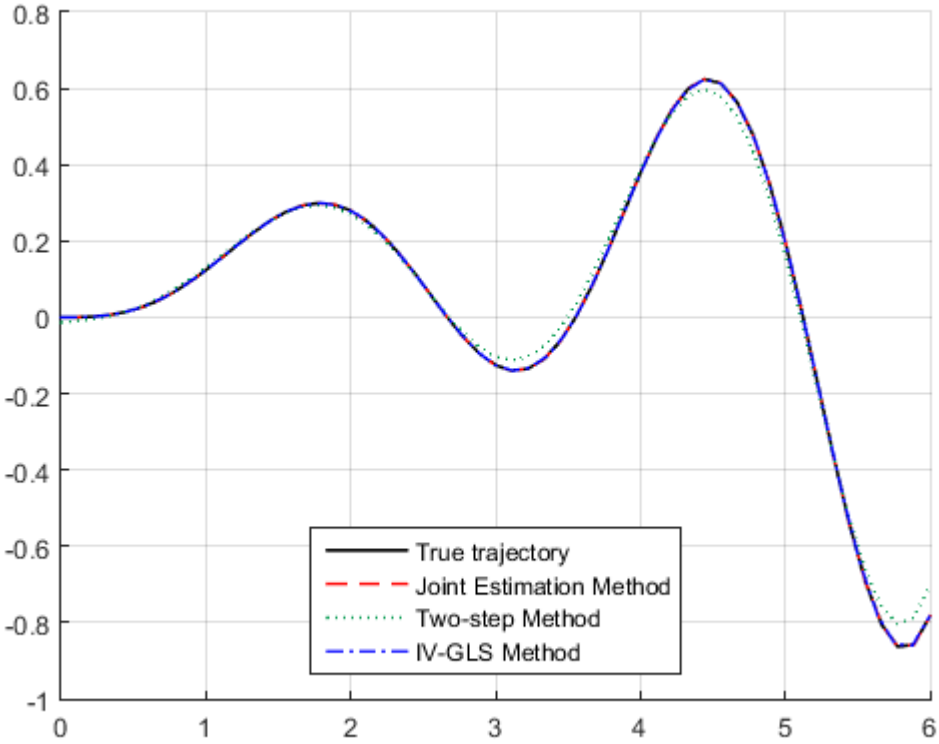

Figure 5.1 True and reconstructed output trajectories of the system using Joint Estimation, Two-step, and IV-GLS methods with N=50

5.2 AWGN with $\mu = 0$ and $\sigma = 0.3$

After evaluating the noiseless case, we set out to evaluate the method's performance against a system with AWGN. In this run, we use the same system from (4.2) and on the same time interval $[0, 6]$, but with AWGN that has $\mu = 0$ and $\sigma = 0.3$ to emulate noise-perturbed output measurement. Moreover, we increase the number of sample points (N) in this run from 50 to 300. This is because all the methods require more sample points when the system is noisy. The 300 points were sampled from a uniform distribution over $[0, 6]$. Finally, we leverage the same performance measures (The L2 norm and the Infinity norm) to assess the efficiency of the methods.

Table 5.2 shows the estimated parameters and the norm values for each method with the additive noise for this run. The results show that all the methods provide estimated parameter values that are close to the actual parameters of the system. Nevertheless, the IV-GLS method proves to be the most accurate in its estimation, followed by the Joint Estimation and Two-step methods. The results also show that the Joint Estimation method was the most computationally expensive, taking 1 hour and 56 minutes to generate its estimations.

Figure 5.2 shows the noisy and true trajectories of the system compared to the estimated trajectories from three methods. To display the difference between the estimated trajectories more clearly, we exclude the noisy signal from the figure but keep the remaining trajectories in Figure 5.3. It can be observed from the figures that the estimated trajectory from the IV-GLS method is the closest to the true trajectory, which confirms the results from Table 5.2.

Method	**a0**	**a1**	**a2**	**a3**	**Computation time**	**L2 norm**	**Infinity norm**
True parameters	1	5	5	0			
Joint Estimation	1.8419	3.1973	5.5966	-0.1544	1 hour 56 mins 30 s	0.6546	0.0698
Two-step	1.5368	-5.4472	6.3867	-1.6543	52.427 s	0.6486	0.0932
IV-GLS	1.0087	5.0078	4.9996	0.0010	51 mins 2 s	0.6035	0.0139

Table 5.2 Table showing true and approx. par values from a true output with AWGN $\mu = 0$ and $\sigma = 0.3$ and N=300

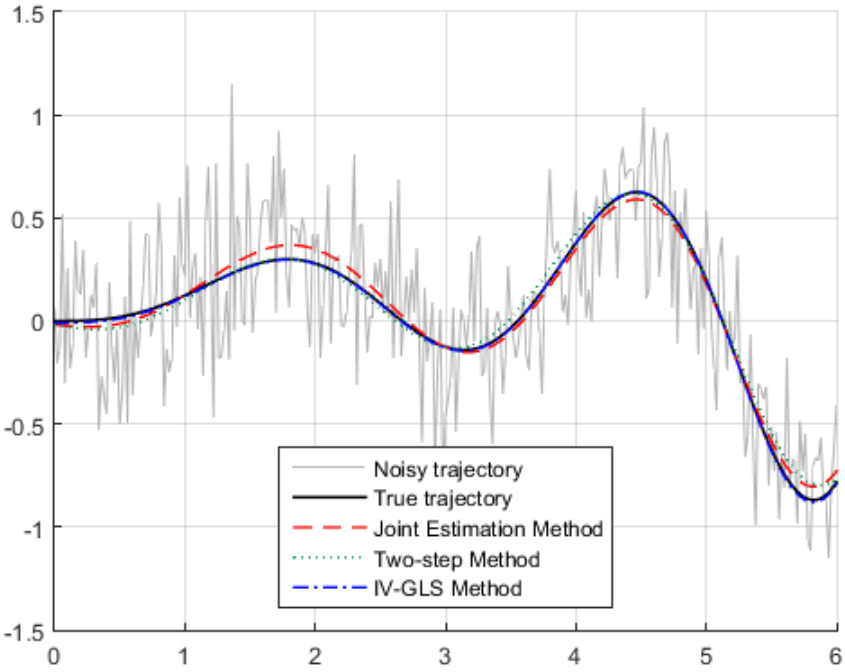

Figure 5.2 Noisy, true, and reconstructed output trajectories of the system with AWGN of $\mu = 0$ and $\sigma = 0.3$ and N=300 using Joint Estimation, Two-step, and IV-GLS methods

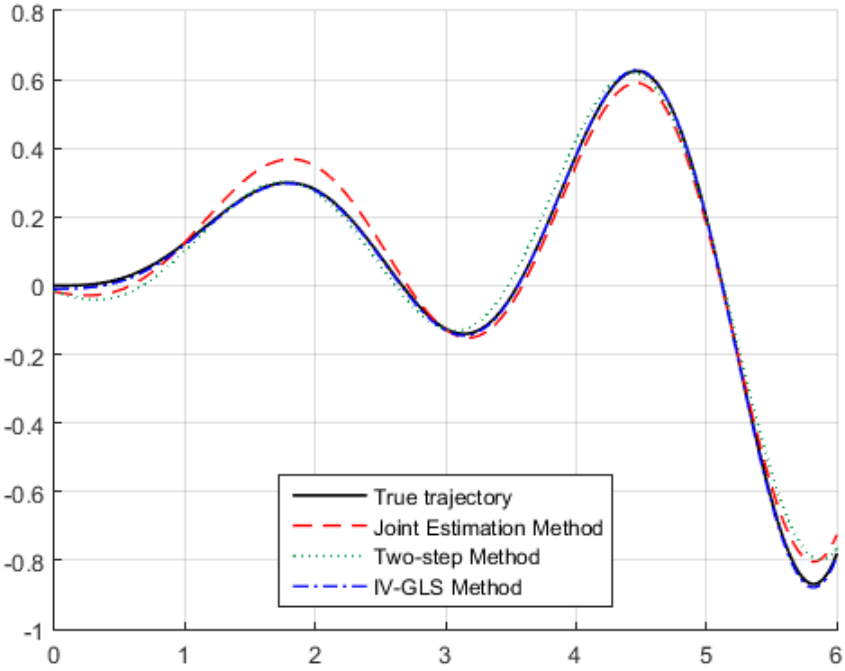

Figure 5.3 True and reconstructed output trajectories of the system with AWGN of $\mu = 0$ and $\sigma = 0.3$ and N=300 using Joint Estimation, Two-step, and IV-GLS methods

5.3 AWGN with $\mu = 0$ and $\sigma = 0.7$

In this run, we increased the standard deviation of the system noise to 0.7. Table 5.3 showcases the resulting parameters and their corresponding computation time, L2 norm and infinity norm values. The results show that, in this run, the IV-GLS outperforms the other methods in the accuracy of estimation. The Two-step method comes second, while the Joint Estimation method produces estimates of the parameters that are significantly different compared to the true values. Computationally, the Joint Estimation method is the most expensive, and it took approximately four times as long compared to the IV-GLS method. Finally, for both the L2 norm and the infinity norm, the IV-GLS method achieves the minimum values.

This is also can be realized from Figure 5.5 where the estimated trajectory by IV-GLS method is almost identical to the true output trajectory. Surprisingly, the Joint Estimation method failed to estimate the trajectory when there is noise with $\mu = 0$ and $\sigma = 0.7$, as it produces a horizontal signal; in other words, it completely fails in reconstructing the trajectory.

Method	**a0**	**a1**	**a2**	**a3**	**Computation time**	**L2 norm**	**Infinity norm**
True parameters	1	5	5	0			
Joint Estimation	131177165.8549	-402536.7984	83351.8333	-6.7451	4 hrs 11 mins 49 s	6.1975	0.8688
Two-step	3.1627	0.7735	5.6422	0.1505	66.689 s	2.7647	0.4217
IV-GLS	1.1856	4.8765	5.1210	-0.0017	1 hr 6 mins 43 s	1.8352	0.0400

Table 5.3 Table showing true and approx. par values from a true output with AWGN $\mu = 0$ and $\sigma = 0.7$ and N=300

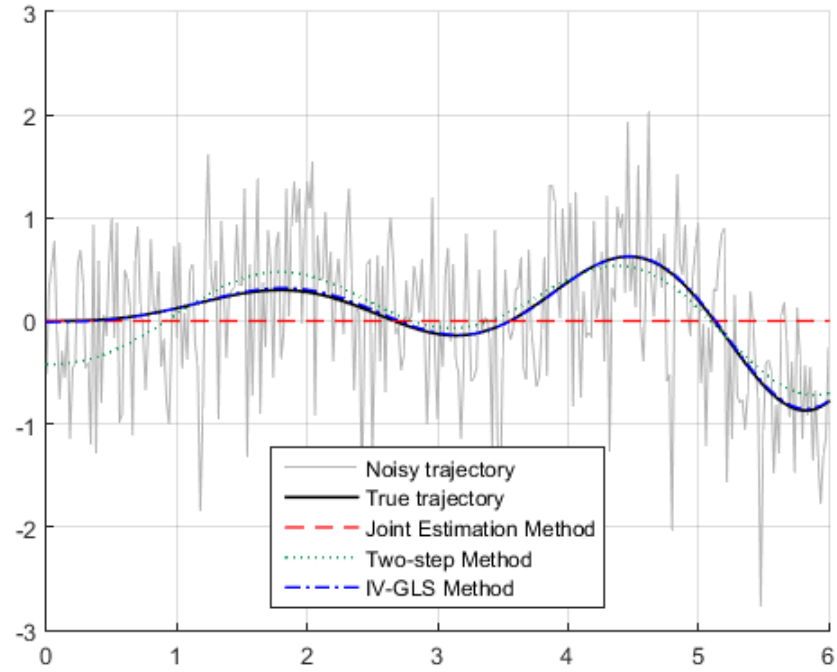

Figure 5.4 Noisy, true, and reconstructed output trajectories of the system with AWGN of $\mu = 0$ and $\sigma = 0.7$ and N=300 using Joint Estimation, Two-step, and IV-GLS methods

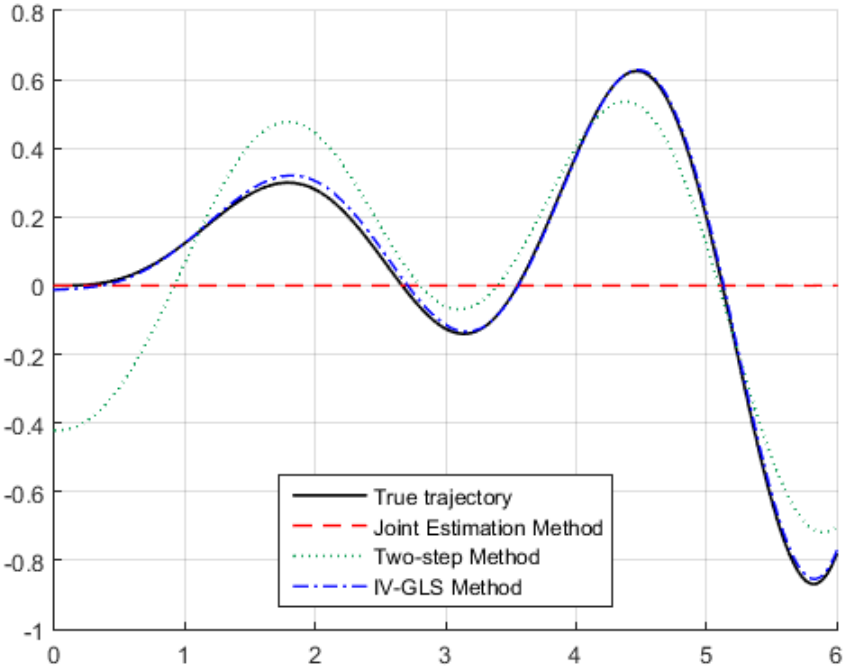

Figure 5.5 True and reconstructed output trajectories of the system with AWGN of $\mu = 0$ and $\sigma = 0.7$ and N=300 using Joint Estimation, Two-step, and IV-GLS methods

5.4 AWGN with $\mu = 0$ and $\sigma = 1$

Last but not least, in this run we increase the noise in the system to have a standard deviation = 1. Similar to the previous runs, we record the methods' estimated parameters and the corresponding performance metrics. The results of this run are shown in Table 5.4. The results show that the IV-GLS is the best performing method in this run, while Two-step comes second with comparable performance. Moreover, the evaluation results show that the difference between the true parameters and the estimated ones by the Joint Estimation method increased compared to the previous run. The Two-step method is the fastest with less than a minute (53 seconds) of execution time, followed by the IV-GLS method which takes a little over 2 hours (2 hrs 10 mins) to finish its computations. The Joint Estimation method takes the longest to execute with a computation time of 24 hrs 59 mins and 37 seconds. Finally, for the norms, the IV-GLS method has minimal values, making it the method with the most accurate estimation.

Figure 5.6 and Figure 5.7 shows the reconstructed trajectories using the three methods. From the figures, we can conclude that the IV-GLS method has the closest estimated trajectory with the regards to the true trajectory, meaning that it is the most accurate. The figures also show that, while the Two-step method can somehow catch the curve, the Joint Estimation method fails to estimate the trajectory in this run.

Method	**a0**	**a1**	**a2**	**a3**	**Computation time**	**L2 norm**	**Infinity norm**
True parameters	1	5	5	0			
Joint Estimation	1272687520.5444	25109474.1712	247130.7714	127.5858	24 hrs 59 mins 37 s	8.0201	0.8690
Two-step	-0.8861	1.3487	7.09	-0.3881	53 s	3.5768	0.6118
IV-GLS	0.7574	4.6523	4.8791	0.1451	2 hrs 10 mins 50 s	3.2552	0.1019

Table 5.4 Table showing true and approx. par values from a true output with AWGN $\mu = 0$ and $\sigma = 1$ and N=500

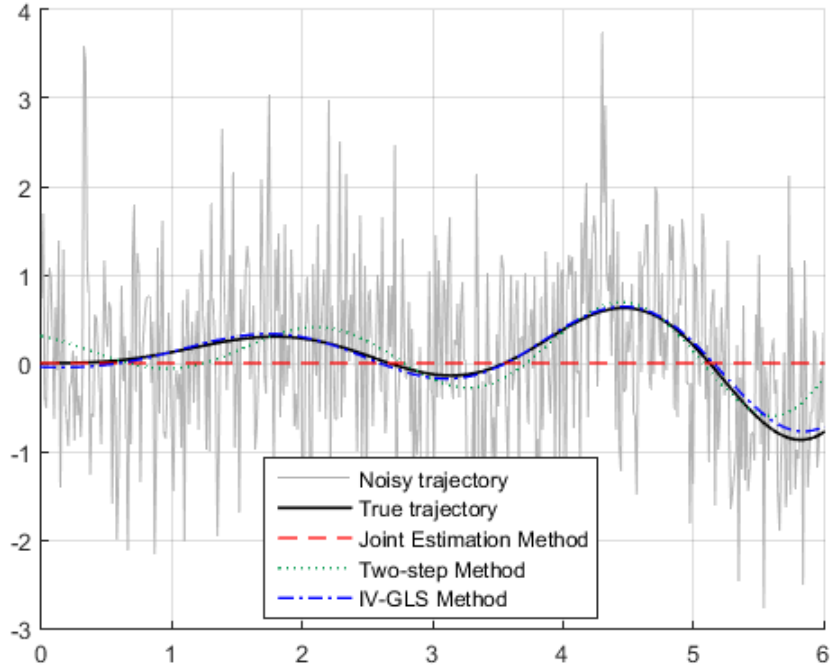

Figure 5.6 Noisy, true, and reconstructed output trajectories of the system with AWGN of $\mu = 0$ and $\sigma = 1$ and N=500 using Joint Estimation, Two-step, and IV-GLS methods

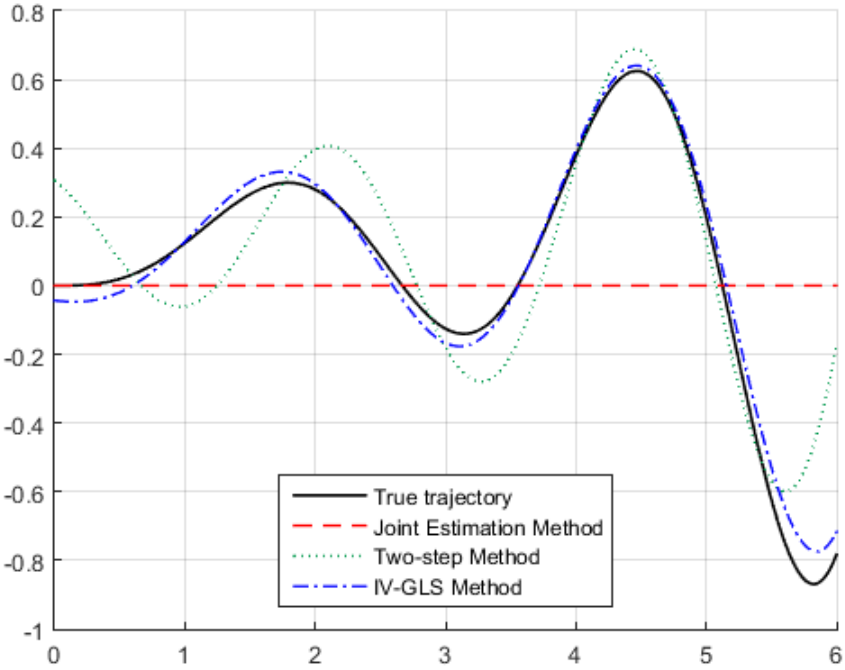

Figure 5.7 True and reconstructed output trajectories of the system with AWGN of $\mu = 0$ and $\sigma = 1$ and N=500 using Joint Estimation, Two-step, and IV-GLS methods

5.5 AWGN with $\mu = 0$ and $\sigma = 1.5$

In our last run, we maximized the AWGN in the system to have a $\sigma = 1.5$. The same evaluateion criteria from the previous runs are used here. The results of this run are shown in Table 5.5, where we can observe that the estimations of the IV-GLS method are the closest to the true parameter values. The Two-step again follows, with estimation that are also not too far from the true parameters. Finally, the results show that the Joint Estimation method is unable to provide an estimate for this run. As for computation time, the Two-step method comes first, followed by the IV-GLS, and Joint Estimation method. The IV-GLS method achives the minimum value for the infinity norm, in this run, making it the most accurate method. However, for estimating the trajectory states, the Two-step method is more efficient as it achieves the minimum L2 Norm value.

We plot the noisy, ture and estimated trajectories for this run in Figures 5.8 and 5.9. From the firgures, we can observe that both the IV-GLS and Two-step were able to develop approximately similar trajectories to the true signal. On the other hand, the Joint Estimation method fails to do so.

Method	**a0**	**a1**	**a2**	**a3**	**Computation time**	**L2 norm**	**Infinity norm**
True parameters	1	5	5	0			
Joint Estimation	1490370691.8895	20396.6919	219410.1197	0.0680	20 hrs 16 mins 59 s	8.0201	0.8690
Two-step	-0.8919	0.5769	6.9429	-0.2402	58.4 s	4.4783	0.713
IV-GLS	0.5125	3.9756	6.0137	0.2271	2 hrs 43 mins 32 s	9.0112	0.3019

Table 5.5 Table showing true and approx. par values from a true output with AWGN $\mu = 0$ and $\sigma = 1.5$ and N=500

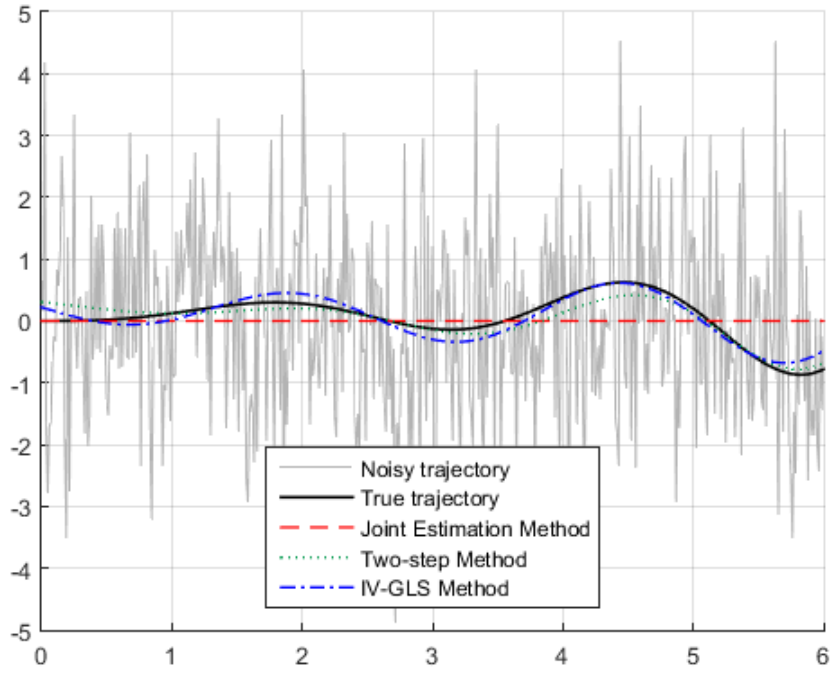

Figure 5.8 Noisy, true, and reconstructed output trajectories of the system with AWGN of $\mu = 0$ and $\sigma = 1.5$ and N=500 using Joint Estimation, Two-step, and IV-GLS methods

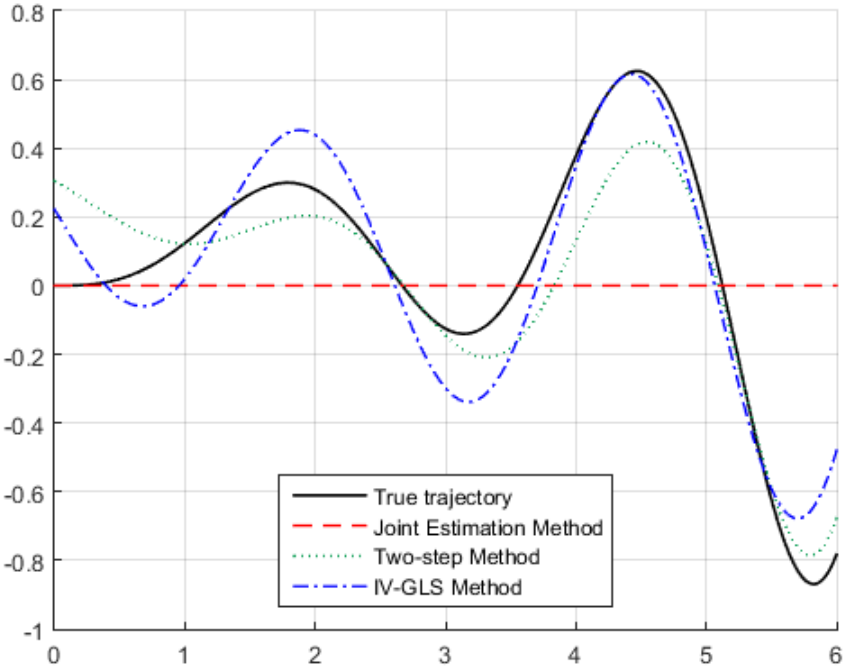

Figure 5.9 True and reconstructed output trajectories of the system with AWGN of $\mu = 0$ and $\sigma = 1.5$ and N=500 using Joint Estimation, Two-step, and IV-GLS methods

5.6 Study of the influence of the number of sampling points

To better understand the effect of increasing the number of sample points on the methods' performance, we run the methods on the same system from (4.2) but with an increased number of samples points. More specifically, we use the aforementioned system on the same time interval $[0, 6]$ and we test the methods using four configurations 1) the system has AWGN with $\mu = 0$ and $\sigma = 1$ and 500 sample points 2) the system has AWGN with $\mu = 0$ and $\sigma = 1$ with 2000 sample points 3) the system has AWGN with $\mu = 0$ and $\sigma = 1.5$ with 500 sample points, and 4) the system has AWGN with $\mu = 0$ and $\sigma = 1.5$ with 3000 sample points. We exclude the Joint Estimation method from this analysis as it is unable to handle large number of sample points (2000 and 3000).

We evaluate the IV-GLS and Two-step methods on all four configurations, and we record the resulting estimated parameter values and corresponding performance metrics. The results of this evaluation are showcased in Table 5.6. From the results, we can observe that both the IV-GLS and Two-step methods produce more accurate estimations with increased sample points. The IV-GLS method, however, produces the most accurate estimated parameters in all four configurations. While the methods' accuracy increases with more sample points, so does the execution time. Both methods take significantly longer time to finish executing when the sample points are increased.

Method	N	σ	a0	a1	a2	a3	Computation	L2	Infinity
True parameters			1	5	5	0	time	norm	norm
Two-step	500	1	-0.8861	1.3487	7.09	-0.3881	53 s	3.5768	0.6118
	2000	1	1.8662	-0.6654	5.6723	-1.1421	4 min 5 s	3.0198	0.2454
IV-GLS	500	1	0.7574	4.6523	4.8791	0.1451	2 hrs 10 mins 50 s	3.2552	0.1019
	2000	1	0.9851	4.9987	5.0066	0.0018	8 hrs 43 mins 20 s	1.9078	0.0685
Two-step	500	1.5	-0.8919	0.5769	6.9429	-0.2402	58.4 s	4.4783	0.713
	3000	1.5	2.6539	-3.1851	6.1401	-0.9916	7 min 10 s	5.3116	0.2098
IV-GLS	500	1.5	0.5125	3.9756	6.0137	0.2271	2 hrs 43 mins 32 s	9.0112	0.3019
	3000	1.5	0.5090	4.2273	5.2360	-0.1905	16 hrs 21 mins 15 s	3.6835	0.1647

Table 5.6 Table showing true and approx. par values from a true output with AWGN $\mu = 0$ and $\sigma = 1$ $\sigma = 1.5$ with varying number of sample points

Figures 5.7 and 5.11 show the true and estimated trajectories of the methods for the system with AWGN $\sigma = 1$ and 500 and 2000 sample points, respectively. We can observe from the figures that the increase in the number of sample points from 500 to 2000 on the same system results in trajectories that are closer to the true trajectory in both the IV-GLS and Two-step methods. For the last two configurations, the results are displayed in Figures 5.9 and 5.13, where, similarly to the first two configurations, both the IV-GLS and Two-step method produce trajectories that better emulate the true trajectory. Nevertheless, in all four figures and configurations, the IV-GLS method is the overall better performing method in terms of accuracy.

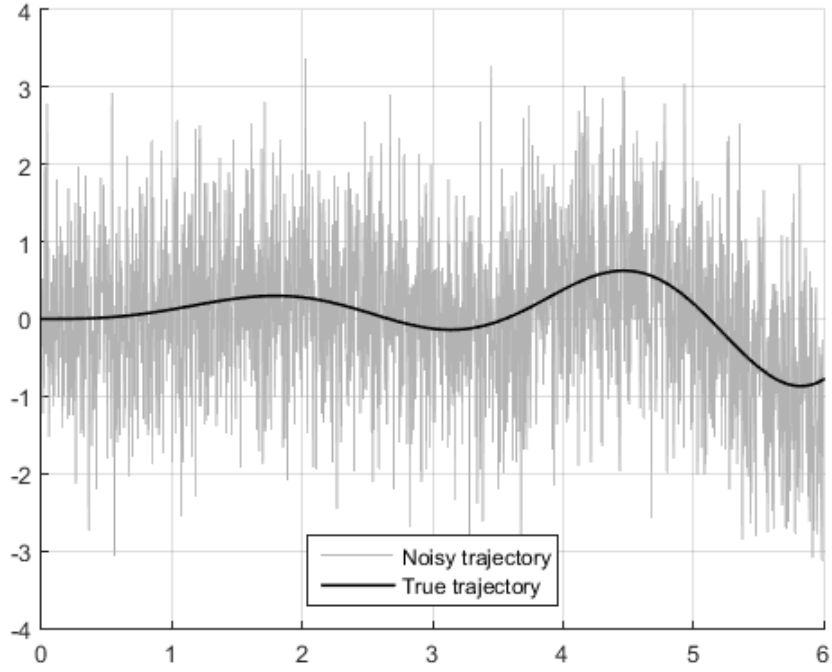

Figure 5.10 Noisy and true output trajectories of the system with AWGN of $\mu = 0$ and $\sigma = 1$ and N=2000

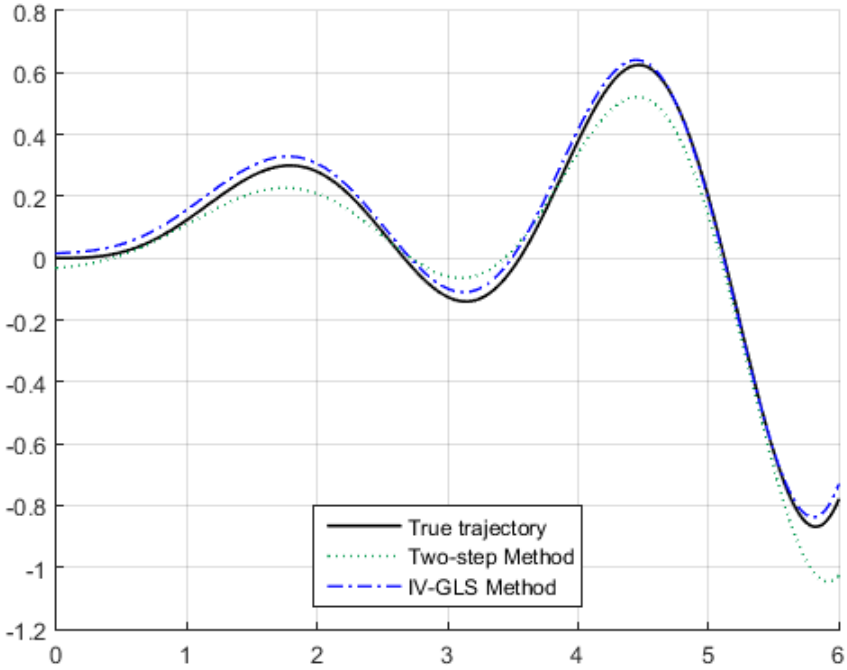

Figure 5.11 True and reconstructed output trajectories of the system with AWGN of $\mu = 0$ and $\sigma = 1$ and N=2000 using Two-step and IV-GLS methods

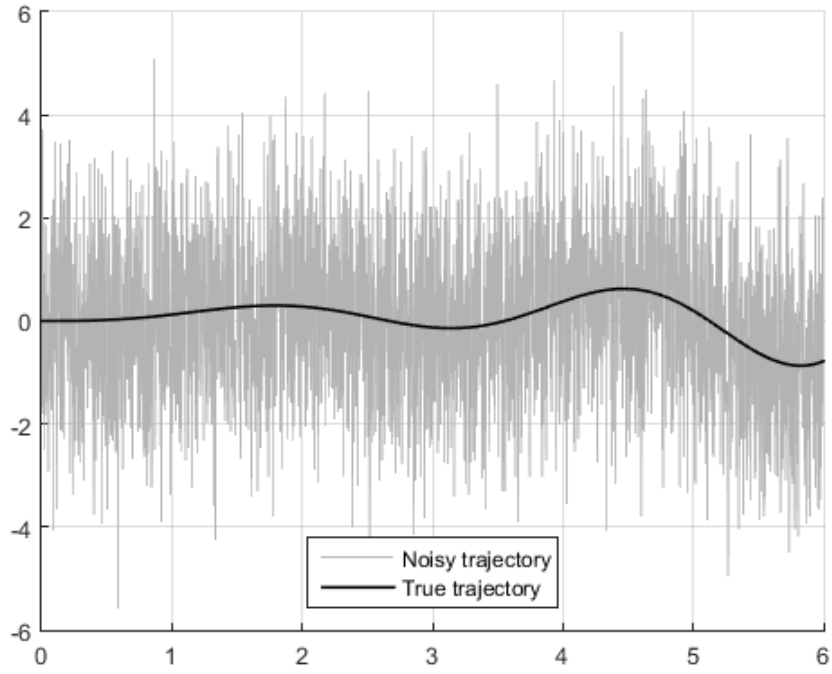

Figure 5.12 Noisy and true output trajectories of the system with AWGN of $\mu = 0$ and $\sigma = 1.5$ and N=3000

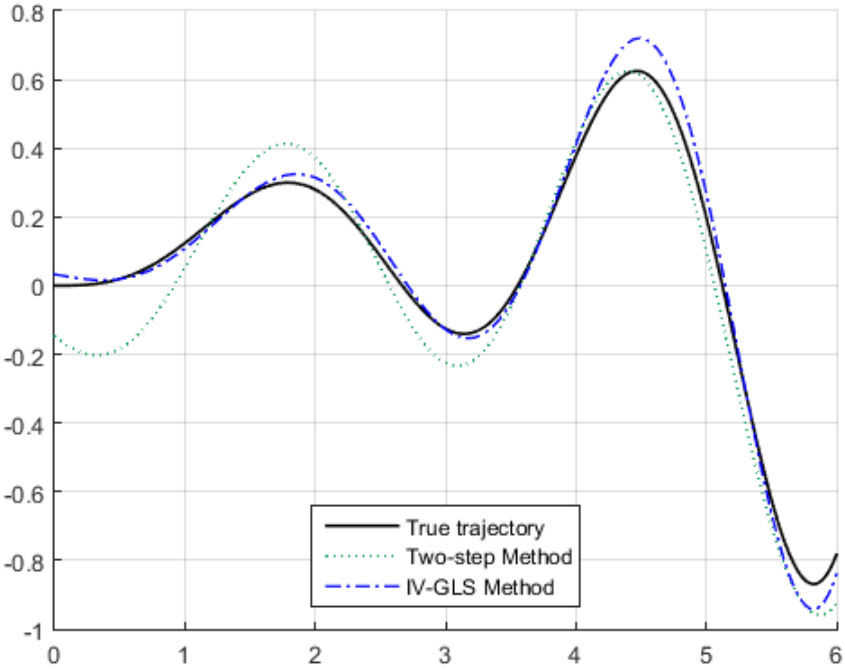

Figure 5.13 True and reconstructed output trajectories of the system with AWGN of $\mu = 0$ and $\sigma = 1.5$ and N=3000 using Two-step and IV-GLS methods

Chapter 6

Discussion

In chapter 5, we showcased our results from the three evaluated methods and for each of the five runs. In this chapter, we deep dive into the results to better understand the methods' performance with no/low noise levels and high noise levels. Finally, we provide an overall (comprehensive) performance metric for all methods.

6.1 No/Low level of Noise

When presented with no/low levels of noise in the system ($\mu = 0$ and $\sigma = 0, 0.3$), all the three evaluated methods are able to produce estimates of the parameters. Moreover, we find that the IV-GLS method generates the most accurate estimates, followed by the Joint Estimation and Two-step methods, respectively. Nevertheless, the Two-step is the fastest in computation time while still producing reasonably accurate estimates.

6.2 High level of Noise

This part of the discussion concerns the methods' performance when presented with systems that have high levels of noise ($\mu = 0$ and $\sigma = 0.7, 1, 1.5$). In these cases, we observe that the

IV-GLS method is the overall best performer when it comes to accuracy, followed by the Two-step method. The Joint Estimation method was unable to obtain sensible estimates in any system with high levels of noise in our experiment. As for computation time, the Two-step method runs significantly faster when compared to the other methods, even when presented with noisy systems. We also studied the effect of increasing the number of sample points in these noisy systems and found that the performance of the IV-GLS and Two-step methods increased when more sample points were provided.

6.3 overall results

Finally, to find out which method is most efficient overall, we leverage an approach used by previous work [1] [33] to give an overall ranking of different methods/techniques based on various performance metrics. More specifically, we give an overall rank to the Joint Estimation method, Two-step method, and IV-GLS method based on their computation time, the L2 Norm and the Infinity Norm results. Table 6.1 shows the results of this approach, where every method is evaluated on each its results in each run. For each run, we rank the methods from 1 to 3 in each metric. For the computation time, the fastest method gets ranked the highest (rank 1), and the slowest method gets the lowest rank (rank 3). Moreover, for the L2 Norm, the lower the value of the L2 norm, the more informative the states of the trajectories are. Hence, lower values of L2 norm are ranked higher. Similarly, smaller infinity norm values correspond to a higher rank in our evaluation.

Finally, after ranking each method in each run, we compute the average for each method's results to obtain an overall ranking for each method.In this step, the method with the lowest average is the highest-ranked and thus the best performing overall. The results of this analysis show that the IV-GLS method has the overall best performance (average rank 1.66) out of the evaluated methods, followed by the Two-step method (average rank 1.86) and the Joint Estimation method (average rank 2.46).

Based on this evaluation, we recommend practitioners to use the IV-GLS method for

general parameter estimation use cases, as it shows the overall best performance, with the highest accuracy and an acceptable speed. Moreover, the IV-GLS method maintains its high performance even when high noise levels are present in the system.

Case	Metric	Joint Estimation	Two step	IV-GLS
Noiseless case	Computation time	1	2	3
	L2 Norm	1	3	2
	Infinity Norm	1	3	2
AWGN with $\mu = 0$ and $\sigma = 0.3$	Computation time	3	1	2
	L2 Norm	3	2	1
	Infinity Norm	2	3	1
AWGN with $\mu = 0$ and $\sigma = 0.7$	Computation time	3	1	2
	L2 Norm	3	2	1
	Infinity Norm	3	2	1
AWGN with $\mu = 0$ and $\sigma = 1$	Computation time	3	1	2
	L2 Norm	3	2	1
	Infinity Norm	3	2	1
AWGN with $\mu = 0$ and $\sigma = 1.5$	Computation time	3	1	2
	L2 Norm	2	1	3
	Infinity Norm	3	2	1
Average (Overall) rank		2.46	1.86	1.66

Table 6.1 Overall ranking of the evaluated methods

Chapter 7

Conclusion and Future Work

Dynamical systems exist in nearly every field, and they can be modelled by ODEs. Some ODEs can not be solved via numerical or analytical methods. In such cases, practitioners leverage parameter estimation methods to estimate the ODE parameters and predict the system's low of motion. However, practitioners face a challenge when selecting the best parameter estimation method to use due to the variety of available methods that they can choose from. Therefore, we compare the performance of four cutting-edge parameter estimation methods to highlight which ones provide optimal behaviour under various conditions of noise and sample sizes.

The evaluation results show that IV-GLS method is the overall best performing method when it comes to the estimation accuracy. Moreover, we find that IV-GLS method can provide an estimation of the parameters for all levels of system noise used in our evaluation. As for the computation time, our results show that Two-step method requires the shortest computation time compared to other methods. Nevertheless, the results show that increasing the number of sample points leads to more accurate estimations by all of the evaluated methods, especially when noise is present in the system.

As future work, we plan to compare more parameter estimation methods with those evaluated in this thesis. More specifically, we plan to evaluate methods that leverage

machine learning techniques, such as deep learning, in their implementation. Moreover, we plan to include more systems from different domains and real-life applications to be used in the evaluation of the methods. These new systems can have different features, such as but not limited to being non-linear systems.

Bibliography

[1] Ahmad Abdellatif, Khaled Badran, Diego Costa, and Emad Shihab. A comparison of natural language understanding platforms for chatbots in software engineering, 12 2020.

[2] Alexander C Aitken. On least squares and linear combination of observations. *Proceedings of the Royal Society of Edinburgh*, 55:42–48, 1936.

[3] Y. Bard. *Nonlinear parameter estimation*. Academic Press, 1974.

[4] L. T. Biegler, J. J. Damiano, and G. E. Blau. Nonlinear parameter estimation: A case study comparison. *AIChE Journal*, 32(1):29–45, 1986. doi: 10.1002/aic.690320105.

[5] Carl De Boor. On calculating with b-splines. *Journal of Approximation Theory*, 6(1): 50–62, 1972. doi: 10.1016/0021-9045(72)90080-9.

[6] Jiguo Cao and James O. Ramsay. Parameter cascades and profiling in functional data analysis. *Computational Statistics*, 22(3):335–351, 2007. doi: 10.1007/s00180-007-0044-1.

[7] Michelle Carey. *Generalised Smoothing in Functional Data Analysis*. PhD thesis, 2012.

[8] Michelle Carey, Eugene G. Gath, and Kevin Hayes. A generalized smoother for linear ordinary differential equations. *Journal of Computational and Graphical Statistics*, 26 (3):671–681, 2017. doi: 10.1080/10618600.2016.1265526.

[9] M. Emre Celebi, Fatih Celiker, and Hassan A. Kingravi. On euclidean norm approximations. *Pattern Recognition*, 44(2):278–283, 2011. doi: 10.1016/j.patcog.2010.08.028.

[10] Vivek Dua. An artificial neural network approximation based decomposition approach for parameter estimation of system of ordinary differential equations. *Computers Chemical Engineering*, 35(3):545–553, 2011. doi: 10.1016/j.compchemeng.2010.06.005.

[11] Shlomo Engelberg. *Random signals and noise: a mathematical introduction*. CRC Press, 2007.

[12] Michel Fliess, Cédric Join, and Hebertt Sira-Ramírez. Non-linear estimation is easy. *International Journal of Modelling, Identification and Control*, 4(1):12–27, 2008.

[13] Andrew Gelman, Frederic Bois, and Jiming Jiang. Physiological pharmacokinetic analysis using population modeling and informative prior distributions. *Journal of the American Statistical Association*, 91(436):1400–1412, 1996. doi: 10.1080/01621459.1996.10476708.

[14] Debarshi Patanjali Ghoshal and Hannah Michalska. Finite-interval kernel-based identification and state estimation for LTI systems with noisy output data. In *2019 American Control Conference (ACC)*, pages 4982–4989. IEEE, 2019.

[15] Debarshi Patanjali Ghoshal and Hannah Michalska. Finite interval estimation of lti systems using differential invariance, instrumental variables, and covariance weighting*. *2020 American Control Conference (ACC)*, 2020. doi: 10.23919/acc45564.2020.9147389.

[16] Debarshi Patanjali Ghoshal, Kumar Gopalakrishnan, and Hannah Michalska. Using invariance to extract signal from noise. In *American Control Conference (ACC), 2017*, pages 2588–2593. IEEE, 2017.

[17] Debarshi Patanjali Ghoshal, Kumar Gopalakrishnan, and Hannah Michalska. Algebraic parameter estimation using kernel representation of linear systems. *IFAC-PapersOnLine*, 50(1):12898–12904, 2017.

[18] Nancy E. Heckman and James O. Ramsay. Penalized regression with model-based penalties. *Canadian Journal of Statistics*, 28(2):241–258, 2000. doi: 10.2307/3315976.

[19] Andrew B. Holder and Marianito R. Rodrigo. An integration-based method for estimating parameters in a system of differential equations. *Applied Mathematics and Computation*, 219(18):9700–9708, 2013. doi: 10.1016/j.amc.2013.03.052.

[20] Yangxin Huang and Hulin Wu. A bayesian approach for estimating antiviral efficacy in hiv dynamic models. *Journal of Applied Statistics*, 33(2):155–174, 2006. doi: 10.1080/02664760500250552.

[21] Yangxin Huang, Dacheng Liu, and Hulin Wu. Hierarchical bayesian methods for estimation of parameters in a longitudinal hiv dynamic system. *Biometrics*, 62(2): 413–423, 2006. doi: 10.1111/j.1541-0420.2005.00447.x.

[22] Anju John. Estimation for siso lti systems using differential invariance. Master's thesis, 2019.

[23] Hua Liang and Hulin Wu. Parameter estimation for differential equation models using a framework of measurement error in regression models. *Journal of the American Statistical Association*, 103(484):1570–1583, 2008. doi: 10.1198/016214508000000797.

[24] Thomas E. Marlin. *Process control: designing processes and control systems for dynamic perfomance*. McGraw-Hill, 2000.

[25] Hannah Michalska. Instructor Notes: Parameter Estimation using Lsplines in RKHS, 2019.

[26] M. Peifer and J. Timmer. Parameter estimation in ordinary differential equations for biochemical processes using the method of multiple shooting. *IET Systems Biology*, 1 (2):78–88, 2007. doi: 10.1049/iet-syb:20060067.

[27] A.a. Poyton, M.s. Varziri, K.b. Mcauley, P.j. Mclellan, and J.o. Ramsay. Parameter estimation in continuous-time dynamic models using principal differential analysis. *Computers Chemical Engineering*, 30(4):698–708, 2006. doi: 10.1016/j.compchemeng.2005.11.008.

[28] H. Putter, S. H. Heisterkamp, J. M. A. Lange, and F. De Wolf. A bayesian approach to parameter estimation in hiv dynamical models. *Statistics in Medicine*, 21(15): 2199–2214, 2002. doi: 10.1002/sim.1211.

[29] J. O. Ramsay. Principal differential analysis: Data reduction by differential operators. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(3):495–508, 1996. doi: 10.1111/j.2517-6161.1996.tb02096.x.

[30] J. O. Ramsay and B. W. Silverman. *Functional data analysis*. Springer, 2 edition, 2005.

[31] J. O. Ramsay, G. Hooker, D. Campbell, and J. Cao. Parameter estimation for differential equations: a generalized smoothing approach. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 69(5):741–796, 2007. doi: 10.1111/j.1467-9868.2007.00610.x.

[32] Stefan Schäffler. *Generalized Stochastic Processes: Modelling and Applications of Noise Processes*. Springer, 2018.

[33] Emad Shihab, Zhen Ming Jiang, Bram Adams, Ahmed E. Hassan, and Robert Bowerman. Prioritizing the creation of unit tests in legacy software systems. *Software: Practice and Experience*, 41(10):1027–1048, 2011. doi: 10.1002/spe.1053.

[34] Hebertt Sira-Ramírez, Carlos García Rodríguez, John Cortés Romero, and Alberto Luviano Juárez. *Algebraic identification and estimation methods in feedback control systems*. John Wiley & Sons, 2014.

[35] Deepak Sridhar, Debarshi Patanjali Ghoshal, and Hannah Michalska. B-splines in joint parameter and state estimation in linear time-varying systems. In *2018 Annual American Control Conference (ACC)*, pages 3508–3513. IEEE, 2018.

[36] J. M. Varah. A spline least squares method for numerical parameter estimation in differential equations. *SIAM Journal on Scientific and Statistical Computing*, 3(1): 28–46, 1982. doi: 10.1137/0903003.

[37] Alan E. Willner and Alan E. Willner. *Chapter 20 - Information capacity of optical channels*. Academic Press is an imprint of Elsevier, 2020.

[38] Peter C Young. *Recursive estimation and time-series analysis: An introduction for the student and practitioner*. Springer Science & Business Media, 2011.