An Investigation of Alternative Techniques for Parameter Estimation Problems for Systems of ODEs

by

Jienan Yao

A project submitted in conformity with the requirements
for the degree of Master of Science
Graduate Department of Computer Science
University of Toronto

Abstract

An Investigation of Alternative Techniques for Parameter Estimation Problems for Systems of ODEs

Jienan Yao
Master of Science
Graduate Department of Computer Science
University of Toronto
2020

Estimating parameters for system of ODEs that best fit observed data can be challenging. The evaluation of the objective function for the parameter estimation usually involves numerically simulating the system of ODEs, which can be computationally expensive and may even suffer from simulation failure. Previous investigation suggested the use of a two-stage procedure by first determining promising candidate parameters using a related objective function that is easy to evaluate before applying a gradient-based optimizer using the full simulation objective function. In this investigation, techniques are considered for the first stage in the two-stage procedure for the parameter estimation problem. These techniques include physics-informed neural networks, Gaussian process regression, Koopman-based lifting methods and unscented Kalman filtering. Other modifications to these techniques that improve the initial guesses for a better chance of converging to the global optimum are also investigated. Experimental results for a set of test problems from the literature are included for comparison between these approaches.

Acknowledgements

I would like to express my sincere gratitude to Prof. Wayne Enright, Prof. David Duvenaud and Jonathan Calver during my MSc program. I am grateful for all their efforts, guidance and support. It was my great fortune to have their help from the enlightening discussions and valuable suggestions during the research project.

Contents

Chapter 1

Introduction

Ordinary Differential Equations (ODEs) are commonly used in mathematical modelling in several application areas such as investigations of population dynamics [2], enzyme kinetics [14], cell signaling and biochemical pathways [21, 25], etc. Parameters are usually contained in these models for them to be fully specified, and in addition, initial conditions for the state variables are specified to form an Initial Value Problem (IVP).

A parameter estimation problem for a system of ODEs arises when we use observed experimental data and seek to find suitable values for the parameters that best fit the model, in the sense of seeking parameters to minimize the discrepancy between the simulated trajectory using the estimated parameters and the observed data, as measured by a specified objective function. The evaluation of the objective function involves numerically simulating the trajectory of the model to compare the simulated state variables and the observed data. This poses an additional challenge compared to other nonlinear optimization problems, since the numerical simulation required to integrate from the initial state until the end of the time interval of interest is usually computationally expensive and may even suffer from occasional failures. Therefore, investigation of the numerical techniques that are efficient yet reasonably accurate is needed for parameter estimation of ODEs.

We first present the formulation of an ODE parameter estimation problem. We consider the parameterized (IVP),

$$y'(t) = f(t, y(t), p),\ y(0) = y_0,\ t \in [0, T], \tag{1.1}$$

where $y \in \mathbb{R}^{n_y}$.

Given a collection of observations $\{\hat{y}(t_i)\}_{i=1}^{n_o}$, we want to find a vector of parameters

$p \in \mathbb{R}^{n_p}$ such that p minimizes the following least squares objective function,

$$L(p) = \sum_{i=1}^{n_o} \sum_{j=1}^{n_y} \frac{(\hat{y}_j(t_i) - y_j(t_i, p))^2}{2\sigma_{ij}^2}. \tag{1.2}$$

We assume that the noise is normally distributed as $\mathcal{N}(0, \sigma_{ij}^2)$ with variance σ_{ij}^2, and assume that time is measured so that time measurements $\{(t_i)\}_{i=1}^{n_o}$ are accurate and do not affect the numerical results we obtain. This allows us to formulate the problem of minimizing (1.2) as an ordinary least-square problem. There have been several techniques proposed to solve the above optimization problem of minimizing (1.2), and a few of them are selected for conducting numerical experiments. Later in this investigation, a total least-square approach will be considered to reflect the noise assumption in the measurement data. And we may also further allow collecting measurements over a more general distribution of the time points other than the uniform time point grids.

We consider a class of problems outlined in Section (2) that have for example been proposed in various modelling tasks appearing in the literature, and experiments are conducted over these well-known test problems using the selected techniques. The problems include the Barnes problem previously investigated in [32, 33], the Goodwin Problem [14], the Mendes Problem [25] and the more challenging Calcium Ion problem [21]. These test problems are also investigated in [5] and we provide background information adopted from [5] in Section (2). In Section (3), detailed descriptions of the techniques considered will be presented, with some of the test problems taken as illustrating examples. We include the numerical experiment results in Section (4) and provide discussion in Section (5).

Chapter 2

Systems of ODEs in the Test Problems

2.1 Calcium Ion Problem

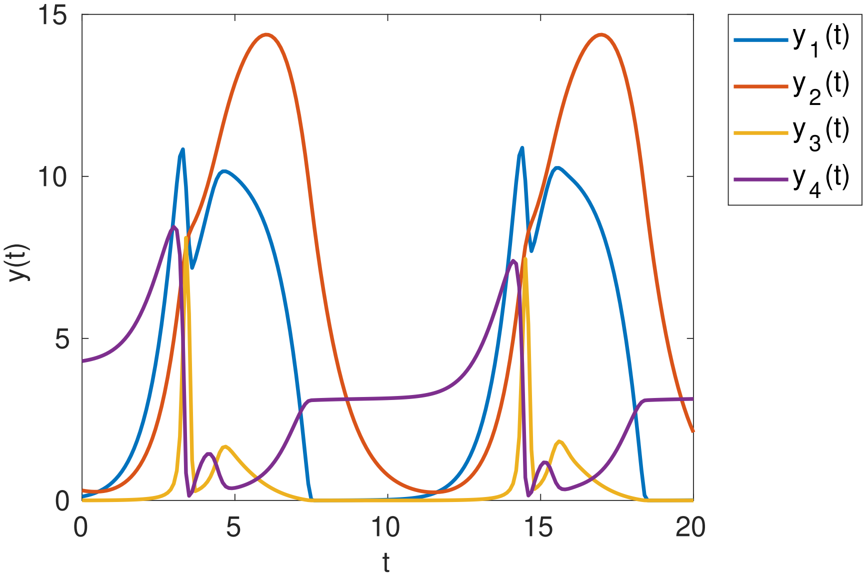

Figure 2.1: Plot of the true trajectory for the Calcium Ion test problem.

This system of ODEs describes the oscillations of Ca^{2+} ions in the cytoplasm of eukaryotic cells, which play a role in cellular information processing. For a complete description of this model, see [21] where this model was first proposed. The model, as originally specified, is given by,

$$
\begin{aligned}
{G^*}_\alpha{}' &= k_1 + k_2 {G^*}_\alpha - k_3 PLC^* \frac{{G^*}_\alpha}{{G^*}_\alpha + Km_1} - k_4 Ca_{cyt} \frac{{G^*}_\alpha}{{G^*}_\alpha + Km_2}, && (2.1)\\
PLC^{*\prime} &= k_5 {G^*}_\alpha - k_6 \frac{PLC^*}{PLC^* + Km_3}, && (2.2)\\
{Ca_{cyt}}' &= k_7 PLC^* Ca_{cyt} \frac{Ca_{er}}{Ca_{er} + Km_4} + k_8 PLC^* + k_9 {G^*}_\alpha \\
&\quad - k_{10} \frac{Ca_{cyt}}{Ca_{cyt} + Km_5} - k_{11} \frac{Ca_{cyt}}{Ca_{cyt} + Km_6}, && (2.3)\\
{Ca_{er}}' &= -k_7 PLC^* Ca_{cyt} \frac{Ca_{er}}{Ca_{er} + Km_4} + k_{11} \frac{Ca_{cyt}}{Ca_{cyt} + Km_6}, && (2.4)
\end{aligned}
$$

where the state variables are concentrations of four compounds, which interact in the calcium-signaling pathway. For consistency with our notation, let ${G^*}_\alpha = y_1$, $PLC^* = y_2$, $Ca_{cyt} = y_3$, $Ca_{er} = y_4$, then the right-hand-side $f(t, y(t), p)$ of (1.1) can be written as $[f_1, f_2, f_3, f_4]^T$. The true parameters for the problem are chosen to be $k_1 = 0.09$, $k_2 = 2$, $k_3 = 1.27$, $k_4 = 3.73$, $k_5 = 1.27$, $k_6 = 32.24$, $k_7 = 2$, $k_8 = 0.05$, $k_9 = 13.58$, $k_{10} = 153$, $k_{11} = 4.85$, $Km_1 = 0.19$, $Km_2 = 0.73$, $Km_3 = 29.09$, $Km_4 = 2.67$, $Km_5 = 0.16$, $Km_6 = 0.05$. Initial conditions for generating the true underlying trajectory are given by $y_1(0) = 0.12$, $y_2(0) = 0.31$, $y_3(0) = 0.0058$, $y_4(0) = 4.3$, and they are assumed to be known when estimating the other parameters. In this test problem, only linear parameters are to be estimated and all of the 6 nonlinear ones are considered as fixed, resulting in a total of 11 linear parameters to be estimated. The true trajectory is generated over the time interval $[0, 20]$, and a total of 201 uniformly spaced observation points along the trajectory are taken every 0.1 time unit ($\Delta t = 0.1$). See Figure 2.1 for the plot of the true trajectory corresponding to the choice of true parameters and initial conditions. Noise is added relative to the magnitude of each component of the state variables and we corrupt the true state variables with Gaussian noise whose standard deviation is proportional to 6.5% of the magnitude of the state variables, i.e, $\sigma_{ij} = 0.065 \cdot y_j(t_i, p_{true})$, where p_{true} is the vector of true parameter values.

2.2 Barnes Problem

The Barnes Problem is often used in the parameter estimation literature for ODE models [32, 33]. It refers to a specific parameterization of the predator-prey model, given by,

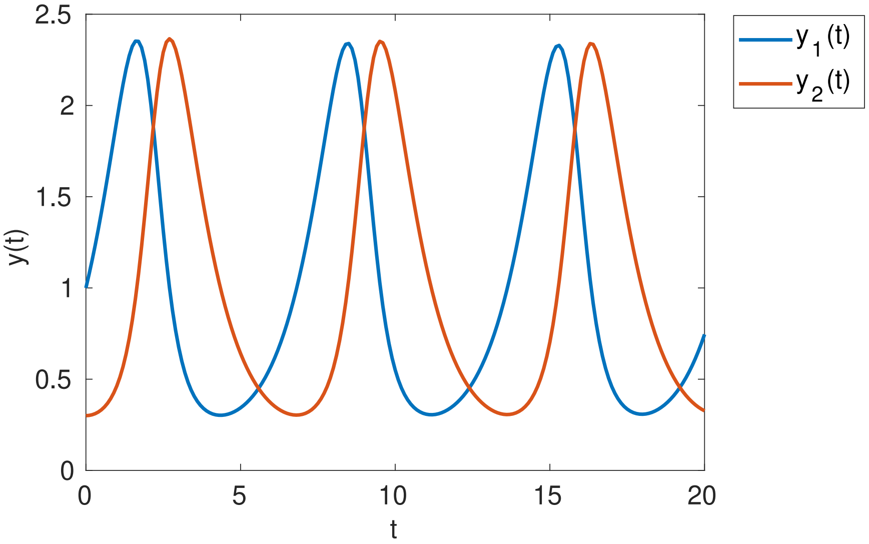

Figure 2.2: Plot of the true trajectory for the Barnes test problem.

$$y_1'(t) = ay_1(t) - by_1(t)y_2(t) \tag{2.5}$$

$$y_2'(t) = by_1(t)y_2(t) - cy_1(t) \tag{2.6}$$

where $y_1(t)$ is the population of predators and $y_2(t)$ is the population of prey. The true parameter values are given by $a = 1$, $b = 1$, $c = 1$, and the initial conditions are given by $y_1(0) = 1$ and $y_2(0) = 0.3$. The true trajectory is simulated over the time interval $[0, 20]$, with a total of 41 uniformly spaced observation points taken every 0.5 time units. See Figure 2.2 for the plot of the true trajectory corresponding to the choice of true parameters and initial conditions. Gaussian noise relative to 10% of the magnitude of each component of the true state variables is added to finally generate the corrupted observation datapoints.

2.3 Goodwin Problem

This system of ODEs models a biological oscillator [14], which has been investigated in applications including enzyme kinetics and circadian clocks [13]. This negative feedback loop model is given by,

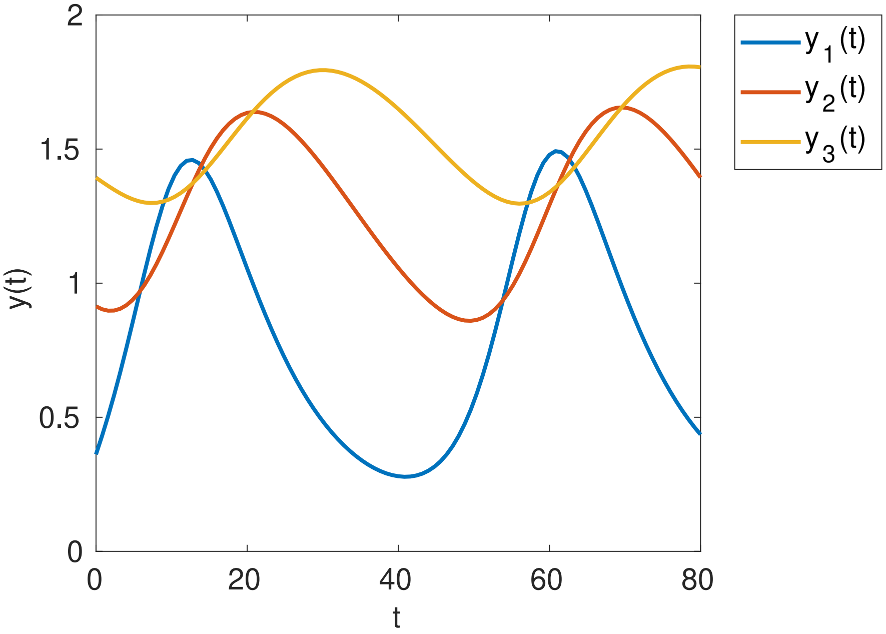

Figure 2.3: Plot of the true trajectory for the Goodwin test problem.

$$y_1'(t) = \frac{a}{A + y_3(t)^\sigma} - by_1(t) \tag{2.7}$$

$$y_2'(t) = \alpha y_1(t) - \beta y_2(t) \tag{2.8}$$

$$y_3'(t) = \gamma y_2(t) - \delta y_3(t) \tag{2.9}$$

There are 6 linear parameters and 2 nonlinear ones, namely A and σ. The true parameters values are given by $a = 3.4884$, $A = 2.15$, $b = 0.0969$, $\alpha = 0.0969$, $\beta = 0.0581$, $\gamma = 0.0969$, $\sigma = 10$, $\delta = 0.0775$ and the initial conditions are given by $y_1(0) = 0.3617$, $y_2(0) = 0.9137$, and $y_3(0) = 1.393$. In our investigations, the nonlinear parameters A and σ are treated to be known and fixed, and only the other 6 linear parameters are to be estimated. The true trajectory is generated over the time interval $[0, 80]$, and a total of 101 uniformly spaced observation points are taken, with $\Delta t = 0.8$. See Figure 2.3 for the plot of the true trajectory corresponding to the choice of true parameters and initial conditions. Gaussian noise with standard deviation proportional to 6.5% of the magnitude of each component of the state vectors is then added to generate the observed datapoints for the parameter estimation task.

2.4 Mendes Problem

Mendes problem is a benchmark problem that was originally posed in [25] and has been subsequently studied [29, 28, 1, 10]. In our notation, this model can be expressed as,

$$y_1'(t) = \frac{k_1}{1+\left(\frac{P}{q_1}\right)^{q_2}+\left(\frac{q_3}{S}\right)^{q_4}} - k_2 y_1 \tag{2.10}$$

$$y_2'(t) = \frac{k_3}{1+\left(\frac{P}{q_5}\right)^{q_6}+\left(\frac{q_7}{y_7}\right)^{q_8}} - k_4 y_2 \tag{2.11}$$

$$y_3'(t) = \frac{k_5}{1+\left(\frac{P}{q_9}\right)^{q_{10}}+\left(\frac{q_{11}}{y_8}\right)^{q_{12}}} - k_6 y_3 \tag{2.12}$$

$$y_4'(t) = \frac{k_7 y_1}{y_1+q_{13}} - k_8 y_4 \tag{2.13}$$

$$y_5'(t) = \frac{k_9 y_2}{y_2+q_{14}} - k_{10} y_5 \tag{2.14}$$

$$y_6'(t) = \frac{k_{11} y_3}{y_3+q_{15}} - k_{12} y_6 \tag{2.15}$$

$$y_7'(t) = \frac{k_{13} y_4 (\frac{1}{q_{16}})(S-y_7)}{1+(\frac{S}{q_{16}})+(\frac{y_7}{q_{17}})} - \frac{k_{14} y_5 (\frac{1}{q_{18}})(y_7-y_8)}{1+(\frac{y_7}{q_{18}})+(\frac{y_8}{q_{19}})} \tag{2.16}$$

$$y_8'(t) = \frac{k_{14} y_5 (\frac{1}{q_{18}})(y_7-y_8)}{1+(\frac{y_7}{q_{18}})+(\frac{y_8}{q_{19}})} - \frac{k_{15} y_6 (\frac{1}{q_{20}})(y_8-P)}{1+(\frac{y_8}{q_{20}})+(\frac{P}{q_{21}})}. \tag{2.17}$$

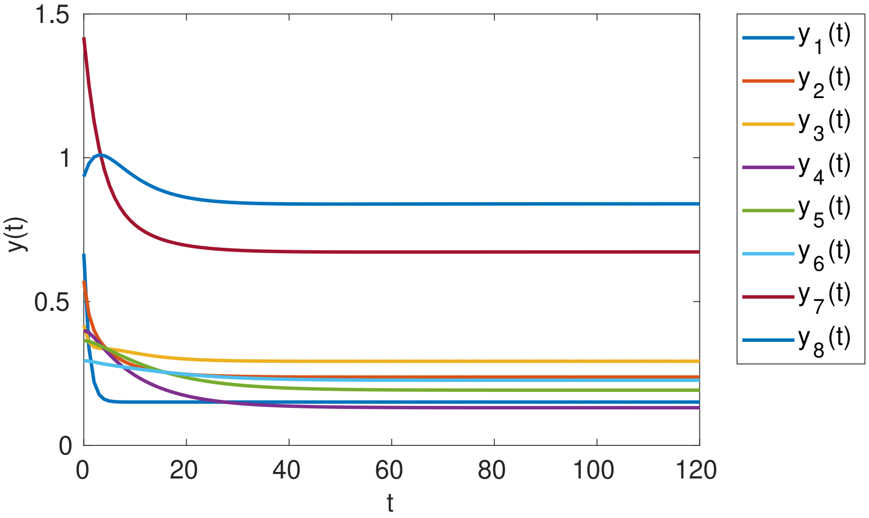

Figure 2.4: Plot of one of the 16 true trajectories in the Mendes test problem, corresponding to $P = 1$, $S = 0.4642$.

There are a total of 36 parameters in the model ($k_1, \ldots, k_{15}$ and $q_1, \ldots, q_{21}$), with 15 linear parameters and 21 nonlinear parameters. Each parameter is assumed to lie in the range $[10^{-12}, 10^6]$, except for the Hill coefficients (q_2, q_4, q_6, q_8, q_{10}, q_{12}), which are assumed to lie in $[0.1, 10]$. The true parameters are given by $k_{1-6} = 1$, $k_{7-12} = 0.1$, $k_{13-15} = 1$, $q_{1,3,5,7,9,13-21} = 1$, $q_{2,4,6,8,10,12} = 2$. Initial conditions are chosen to be,

$$y(0) = [0.66667, 0.57254, 0.41758, 0.4, 0.36409, 0.29457, 1.419, 0.93464],$$

P is chosen from {0.05, 0.13572, 0.3684, 1} and S is chosen from {0.1, 0.46416, 2.1544, 10}. These values of P's and S's together form 16 combinations and simulations are run for each of these combinations. For each of the trajectories corresponding to one pair of P and S, a total of 21 uniformly spaced observation points are taken over the time interval $[0, 120]$ with $\Delta t = 6$, and there are 16 sets of such state variable vector data resulting in a total of $16 \times 21 = 336$ state vector data points. See Figure 2.4 for the plot of one of the 16 true trajectories corresponding to one pair of P and S values with $P = 1$ and $S = 0.4642$. In the test problem, all nonlinear parameters are fixed so that only linear parameters are to be estimated, and initial conditions as well as the values of P's and S's are assumed to be known. Relative Gaussian noise with standard deviation to be 3% of the magnitude of each component of the the state variables is added to generate the observed data.

In addition to the experiments using observed data scattered over uniform grid of timepoints, we also note that two of the techniques we consider INT-SME and SME, are justified by and dependent on a reasonably good approximation to the underlying true trajectory. This indicates that we may need more data points in the neighbourhood of some critical points within region of interest to achieve smoother approximations. Specifically, if the state variable is changing rapidly within certain region of the time interval, it may not be enough to interpolate the observed trajectory well if we only include observations over uniformly spaced timepoints. In order to account for the potential smoothness issue in interpolation, we allow the experiments to include additional observations points over a finer mesh in regions where the state variables change rapidly, and where we expect to achieve more suitable approximation.

Chapter 3

Techniques Considered

Typical gradient-based optimization with direct simulation of IVP (1.1) may suffer from a high cost of the numerical solver used in the simulation, and the risk of failure arising during the simulation. In addition, at best only a local minimum may be obtained if starting with an initial guess not sufficiently close to the true optimum value. The Smooth and Match Estimator (SME) [15] is motivated by the fact that, a good initial guess can often be obtained by introducing a related parameter estimation problem with the objective function,

$$\min_p \int_0^T \|(\tilde{y}'(t) - f(t, \tilde{y}(t), p))\|^2 \, dt, \tag{3.1}$$

where is $\tilde{y}(t)$ a smoothed curve designed to match the observed target trajectory. Equation (3.1) involves only trajectory time derivative estimation on the smoothed curve and a numerical quadrature rule can be used to approximate the value of this objective function instead of a usually more expensive numerical simulation of (1.1). With the objective function (3.1), samples drawn from larger areas of the search space can be considered in an attempt to reduce the chance of being trapped at a local minimum. Also, as will be discussed in the next section, if the IVP (1.1) has some special structure, some of the parameters which appear non-linearly in the objective function (1.2) might become linear parameters when the objective function (3.1) is applied.

A two-stage procedure has been developed to address some of the difficulties that arises in the parameter estimation problems: in the first stage, an initial guess of the parameter p_0 is obtained by solving the optimization problem (3.1). Once a good enough initial guess is found, full evaluation of (1.2) by simulating IVP (1.1) and gradient-based techniques are used in the second stage to potentially achieve superlinear convergence by employing a Newton-type optimization technique applied to (1.2).

3.1 Techniques for Obtaining Suitable Initial Guesses

INT-SME Exploiting ODE Structure

Instead of using the derivative of the smoother as in SME, Dattner [8] suggested an integral form of smooth and match estimator (called INT-SME) which is of the form,

$$\min_{p} \int_0^T \left\| \tilde{y}(t) - \big(y_0 + \int_0^t f(s, \tilde{y}(s), p)\, ds\big) \right\|^2 dt, \tag{3.2}$$

Calver [5] proposed an INT-SME formulation that further exploits special structure of the ODE when there are both linear parameters and nonlinear parameters in the ODE. Specifically, he considered ODEs in which the right hand side function f can be decomposed into,

$$f(t, y(t), p) = G(t, y(t), q)r + \overline{G}(t, y(t), q), \tag{3.3}$$

where r is the vector of parameters appearing linearly and q the vector of parameters appearing nonlinearly in f. Note in the experiments reported here, $\overline{G}$ is simply zero. When the structure in (3.3) is considered, (3.2) becomes

$$\min_{q} \int_0^T \left\| \tilde{y}(t) - \Big(y_0 + \big[\int_0^t G(s, \tilde{y}(s), q)\, ds\big] r(q) + \int_0^t g(s, \tilde{y}(s), q)\, ds\Big) \right\|^2 dt, \tag{3.4}$$

where $r(q)$ is the unique linear least square solution for a given fixed q, therefore (3.4) can be solved with respect to only the nonlinear parameters q. In test problems in which all parameters appear linearly, the initial guess of the parameters can be obtained in one iteration of the associated linear least square problem.

For example, for the Calcium Ion problem we introduced in Section (2), the special structure in the system of ODEs (2.1)-(2.4) allows us to decompose the problem into subproblems by grouping the linear and non-linear parameters according to their appearance in the ODEs. The right-hand-side of the IVP in (1.1), $f(t, y(t), p)$ can be decomposed into 3 independent parts, with $p = r$ for the case of linear parameter estimation, and $p = [r; q]$ if nonlinear parameters are to be estimated, where linear parameters $r = [k_1, \ldots, k_{11}]^T \in \mathbb{R}^{11}$ and nonlinear parameters $q = [Km_1, \ldots, Km_6]^T \in \mathbb{R}^6$. Specifically, there are 4 linear parameters and 2 nonlinear parameters in the first subproblem (namely, k_{1-4} and Km_{1-2}), 2 linear parameters and 1 nonlinear parameter in the second subproblem (k_{5-6} and Km_3), and 5 linear parameters and 3 nonlinear parameters in the third subproblem (k_{7-11} and Km_{4-6}). We refer the reader to [5] for a detailed description of this decomposition of the Calcium Ion problem that exploits the sparsity structure in this system of ODEs.

Once we have the observed data or its associated smoothed trajectory, an INT-SME objective function can be formulated for each of the 3 independent subproblems. And the least-square problem of minimizing (3.4) for each of the subproblems can be solved independently.

Physics Informed Neural Networks (PINNs)

Neural networks have gained increasing popularity in machine learning tasks such as classification and regression, and among the recent advances, Physics informed neural networks [26, 27] utilize a deep learning framework for the approximate solution of nonlinear partial differential equation, as well as model parameter estimation. A neural network consists of input units, output units as well as layers of hidden units and the connections between these units together with nonlinear activation function. Each unit calculates its weighted sum of the outputs passed by units from the previous layer connected to it and outputs its activation function value using the calculated weighted sum, which will then be passed to the next layer. In regression tasks, pairs of input and expected output values are collected and form a *training set*. A neural network is *trained* with respect to the weights in the connections to approximate the true input-output function.

PINNs act as such function approximators in a regression task, which by the universal approximator theorem [16] of neural networks, are capable of approximating general systems with physical laws encoded by differential equations. PINNs are used to approximate the solution of the differential equations, with temporal/spatial variables as inputs, and temporal/spatial derivatives calculated through automatic differentiation. The neural networks are trained to minimize an objective function that enforces the differential equation constraints at a set of specified collocation points $\{t_d^i, x_d^i\}_{i=1}^{N_d}$ as well as at initial and boundary points $\{t_u^i, x_u^i\}_{i=1}^{N_u}$. In particular, in the PDE case, we denote the solution function approximated by the neural network as $u(t,x)$, and $d(t,x) = \frac{\partial u}{\partial t} - f(t,x,u(t,x),p)$ (p assumed to be known) is the residual of the right-hand-side function evaluation and time derivative determined by automatic differentiation. The objective function is set to

$$MSE = MSE_u + MSE_d, \tag{3.5}$$

where

$$MSE_u = \frac{1}{N_u}\sum_{i=1}^{N_u} |u(t_u^i, x_u^i) - u^i|^2,\ MSE_d = \frac{1}{N_d}\sum_{i=1}^{N_d} |d(t_d^i, x_d^i)|^2,$$

and u^i is the solution function value given at the initial/boundary point (t_u^i, x_u^i).

The setup of the neural network for a parameter estimation problem is similar, but with the objective set to minimize simultaneously the difference between the output of the network and the observed data as well as the degree to which the governing differential equation fails to be satisfied at the sampled points. In the specific case of applying the technique to ODE parameter estimation, the solution function approximated by neural network is denoted $u(t)$ with $d(t) = \frac{\partial u}{\partial t} - f(t, u(t), p)$, and the objective function becomes

$$\overline{MSE} = \overline{MSE}_u + \overline{MSE}_d, \tag{3.6}$$

where

$$\overline{MSE}_u = \frac{1}{n_o}\sum_{i=1}^{n_o} |u(t_i) - \hat{y}(t_i)|^2,\ \overline{MSE}_d = \frac{1}{n_o}\sum_{i=1}^{n_o} |d(t_i)|^2,$$

in which $\{t_i, \hat{y}(t_i)\}_{i=1}^{n_o}$ are the collected data and both the shared neural network weights in $u(t)$ and $d(t)$ as well as ODE parameters p are optimized with respect to (3.6).

In the experiments conducted in Section (4), the PINN-approximated trajectory is used as a smoother which is obtained by optimizing neural network parameters with respect to only $\overline{MSE}_u$ and then employed in the INT-SME framework to obtain a suitable initial guess. However, since parameter estimation is provided by PINN as well, those estimated parameters could also be interpreted as an alternative initial guess.

Koopman-based Lifting Techniques

The Koopman-based lifting techniques [23] for nonlinear system identification are based on the Koopman Operator [3, 19], and several extensions to Dynamic Mode Decomposition [37, 22, 38].

It approximates the evolution of nonlinear observable functions of the states by defining and approximating linear infinite-dimensional operators (the Koopman Operator) on the observables. We assume the vector field function $f(t, y(t), p)$ in (1.1) can be expressed as a vector-valued function $\mathrm{F}(y) \in \mathbb{R}^{n_y}$ in the form of a weighted combination of known/prescribed library functions. Specifically we assume, the system to be identified has a simpler structure and can be written as

$$y' = \mathrm{F}(y), y \in \mathbb{R}^{n_y}, \tag{3.7}$$

where the vector field $\mathrm{F}(y)$ is a weighted combination of N_F scalar library functions $h_k(y)$,

$$\mathrm{F}(y) = \sum_{k=1}^{N_F} \mathrm{w}_k h_k(y), \tag{3.8}$$

with $\mathrm{w}_k = [w_k^1, \ldots, w_k^{n_y}]^T \in \mathbb{R}^{n_y}$ to be identified. If we consider separating the vector of parameters p into a vector of linear parameters r and a vector of nonlinear parameters q (see equation (3.3) for definition of r and q), w_k will be composed of elements of r, while in the nonlinear parameter estimation case, the library function h_k could also depend on q. Our investigation currently considers only the linear parameter estimation case. Assume we have collected snapshot pairs of observed state variables,

$$[\hat{y}(t_i), \hat{y}(t_{i+1})] \in \mathbb{R}^{n_y \times 2}, i \in \{1, \ldots, n_o - 1\},$$

based on Koopman theory, nonlinear evolution can be approximated by finite-dimensional projection of the Koopman operator. In the main lifting method proposed in [23], N basis functions $[\mathrm{g}_1(y), \ldots, \mathrm{g}_N(y)]^T \equiv \mathrm{g}(y) \in \mathbb{R}^N$ (treated as observable functions) are chosen to approximate the discrete evolution of (3.7) over uniformly spaced observation time points (with $\Delta t = t_{i+1} - t_i$),

$$\mathrm{g}(\hat{y}(t_{i+1}))^T \approx \mathrm{g}(\hat{y}(t_i))^T \mathrm{U} \tag{3.9}$$

where $\mathrm{U} \in \mathbb{R}^{N \times N}$ and $\hat{y}(t_i)$ is the ith observation state variable in time. If we stack each $\mathrm{g}(\hat{y}(t_i))$, $\mathrm{g}(\hat{y}(t_{i+1}))$ and construct the matrices

$$\mathrm{M}_1 = \begin{pmatrix} \mathrm{g}(\hat{y}(t_1))^T \\ \vdots \\ \mathrm{g}(\hat{y}(t_{n_o-1}))^T \end{pmatrix}, \ \mathrm{M}_2 = \begin{pmatrix} \mathrm{g}(\hat{y}(t_2))^T \\ \vdots \\ \mathrm{g}(\hat{y}(t_{n_o}))^T \end{pmatrix}, \tag{3.10}$$

then the linear least-square solution to equation (3.9) can be written as: $\mathrm{U} \approx \mathrm{M_1}^{\dagger}\mathrm{M}_2$, where † denotes the Moore-Penrose pseudo-inverse.

The matrix U is approximated from the observed data, and a matrix logarithm $\mathrm{L} = \frac{1}{\Delta t} \log \mathrm{U}$ is introduced as the matrix such that $\exp\{\Delta t \mathrm{L}\} = \mathrm{U}$. L is used to reconstruct the coefficients $\{\mathrm{w}_k\}_{k=1}^n$ of the library functions $\{h_k\}_{k=1}^n$. Specifically, with L we have an approximation of the vector field in terms of the basis functions chosen, and to recover the coefficients w_k in (3.8), we adopt the straightforward method of projecting the approximated vector field values at the sampled points by a weighted combination of basis function values onto the space spanned by the library functions. Taking into consideration the sparsity structure of right-hand-side of (1.1) (i.e., not all library functions appear in every component of the function $f(t, y(t), p)$), the space spanned by a subset of library functions is considered for each state variable. Solving the least-square problem arising from this linear regression of projecting basis function space onto library function space will provide an initial guess of the ODE parameters. Note [23] also suggests ways to perform identification of the coefficients $\{\mathrm{w}_k\}_{k=1}^n$ for non-polynomial vector field, but it may not enforce certain sparsity structure (see (3.11) for example coefficients in the Calcium Ion problem below) for the library functions. Other vector field identification techniques suggested in [23, 22] using least-square solution over a set of overdetermined set of equations together with sparsity promoting techniques could be further investigated and is left for future work.

In the above derivation of the methods, state variable observations are assumed to be exact and the matrices M_1 and M_2 contain no noise. However, the methods proposed could also suffer from the difficulty of noise in the observation data and denoising in this case could be more complicated than is the case for ordinary least-squares when obtaining an approximation of U. Noise in matrices M_1 and M_2 come from both the measurements and (non-)linear basis functions: the noise term σ_{ij}^2 in (1.2) can affect M_1 and M_2 either appearing as direct entries (through identity basis functions) or through (non-)linear function transformations. Several denoising techniques have been considered to address this issue of noisy observations in the data. There are three denoising techniques proposed in [9], namely 1) noise-corrected DMD (ncDMD), which includes a direct correction of the bias caused by noise and identified using known noise properties, 2) forward/backward DMD (fbDMD), which combines the DMD results in both forward pass and backward pass, and 3) total least-squares DMD (tlsDMD) that accounts for noise involved in both the previous and the next time step when finding the least-squares solution to (3.9). In the experiments reported here, fbDMD seems to be more effective, thus results are presented only for the fbDMD Koopman-based lifting technique.

To illustrate the usage of the Koopman-based lifting technique, the Calcium Ion problem is used to instantiate some of the key matrices/variables introduced above. The system of ODEs in this problem can be expressed in the form of equation (3.8) as $\mathrm{F}(y) = \sum_{k=1}^{N_F} \mathrm{w}_k h_k(y)$ with $N_F = 11$ and the library functions include monomials of the state variables of degree up to 1, together with 6 non-polynomial terms appearing in the right-hand-side of the system of ODEs (2.1)-(2.4), namely $y_2 \frac{y_1}{y_1+Km_1}$, $y_3 \frac{y_1}{y_1+Km_2}$, $\frac{y_2}{y_2+Km_3}$, $y_2 y_3 \frac{y_4}{y_4+Km_4}$, $\frac{y_3}{y_3+Km_5}$ and $\frac{y_3}{y_3+Km_6}$. The true coefficients $\mathrm{w} = [\mathrm{w}_1, \mathrm{w}_2, \dots, \mathrm{w}_{11}] \in \mathbb{R}^{4\times 11}$ for the vector field in this test problem can then be represented as,

$$\mathrm{w} = \begin{bmatrix} k_2 & \cdot & \cdot & \cdot & k_1 & -k_3 & -k_4 & \cdot & \cdot & \cdot & \cdot \\ k_5 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & -k_6 & \cdot & \cdot & \cdot \\ k_9 & k_8 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & k_7 & -k_{10} & -k_{11} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & -k_7 & \cdot & k_{11} \end{bmatrix}, \tag{3.11}$$

where dots represent zero entries in the matrix.

We chose the basis functions to be the same as the library functions in this case. Then g is a function that maps $\mathbb{R}^4$ to $\mathbb{R}^{11}$, and the matrix $\mathrm{M_1}^T = [\mathrm{g}(\hat{y}(t_1)), \ldots, \mathrm{g}(\hat{y}(t_{n_o-1}))]$ can be expanded as,

$$\mathrm{M_1}^T = \begin{bmatrix} \hat{y}_1(t_1) & \cdots & \hat{y}_1(t_{n_o-1}) \\ \hat{y}_2(t_1) & \cdots & \hat{y}_2(t_{n_o-1}) \\ \hat{y}_3(t_1) & \cdots & \hat{y}_3(t_{n_o-1}) \\ \hat{y}_4(t_1) & \cdots & \hat{y}_4(t_{n_o-1}) \\ 1 & \cdots & 1 \\ \hat{y}_2(t_1)\frac{\hat{y}_1(t_1)}{\hat{y}_1(t_1)+Km_1} & \cdots & \hat{y}_2(t_{n_o-1})\frac{\hat{y}_1(t_{n_o-1})}{\hat{y}_1(t_{n_o-1})+Km_1} \\ \hat{y}_3(t_1)\frac{\hat{y}_1(t_1)}{\hat{y}_1(t_1)+Km_2} & \cdots & \hat{y}_3(t_{n_o-1})\frac{\hat{y}_1(t_{n_o-1})}{\hat{y}_1(t_{n_o-1})+Km_2} \\ \frac{\hat{y}_2(t_1)}{\hat{y}_2(t_1)+Km_3} & \cdots & \frac{\hat{y}_2(t_{n_o-1})}{\hat{y}_2(t_{n_o-1})+Km_3} \\ \hat{y}_2(t_1)\hat{y}_3(t_1)\frac{\hat{y}_4(t_1)}{\hat{y}_4(t_1)+Km_4} & \cdots & \hat{y}_2(t_{n_o-1})\hat{y}_3(t_{n_o-1})\frac{\hat{y}_4(t_{n_o-1})}{\hat{y}_4(t_{n_o-1})+Km_4} \\ \frac{\hat{y}_3(t_1)}{\hat{y}_3(t_1)+Km_5} & \cdots & \frac{\hat{y}_3(t_{n_o-1})}{\hat{y}_3(t_{n_o-1})+Km_5} \\ \frac{\hat{y}_3(t_1)}{\hat{y}_3(t_1)+Km_6} & \cdots & \frac{\hat{y}_3(t_{n_o-1})}{\hat{y}_3(t_{n_o-1})+Km_6} \end{bmatrix}, \tag{3.12}$$

and the matrix M_2 can be populated similarly, with data points collected over the time points $t_2, \ldots, t_{n_o}$. Once an approximation of the discrete evolution is obtained using (3.9) with the populated M_1 and M_2, and the matrix logarithm L is obtained for the approximation of the vector field in terms of the basis functions, we need to reconstruct the coefficients $[\mathrm{w}_1, \mathrm{w}_2, \ldots, \mathrm{w}_{11}]$ by projecting the approximated vector field values at the observed time points using library functions evaluated at the same time points. This projection is performed because of the fact that by solving (3.9), sparsity of the weights w_k's for the vector field as illustrated in (3.11) may not be observed and the approximated coefficients $\widetilde{\mathrm{w}}_k$'s could contain more non-zero entries than desired. In the Calcium Ion problem for the first state variable, the approximated vector field values $\widetilde{\mathrm{F}}_1 = [\widetilde{\mathrm{F}}_1(t_1), \ldots \widetilde{\mathrm{F}}_1(t_{n_o})]^T$ are obtained by taking a weighted combination of all 11 basis functions, which is $[\mathrm{L}_1^T \cdot \mathrm{g}(\hat{y}(t_i)), \ldots, \mathrm{L}_1^T \cdot \mathrm{g}(\hat{y}(t_{n_o}))]$, where L_1 is the column of L related to the basis function y_1. Note the true coefficients for the first state variable only involve 4 non-zero weights, namely, k_1,k_2,k_3 and k_4 (see the first row in (3.11)), while components in L_1 could all be non-zero. We then project the approximated vector field $\widetilde{\mathrm{F}}_1$ in a least-square way, with the 4 library functions involved in (2.1), namely 1, y_1, $y_2(t_1)\frac{y_1}{y_1+Km_1}$ and $y_3\frac{y_1}{y_1+Km_2}$. These approximated vector field values at the observed time points are used as right-hand-side of the least-square problem with the 4 library functions evaluated at the datapoints as the left-hand-side, and we seek least-square solution for $[k_1, k_2, k_3, k_4]^T$ as the approximation for the (non-zero) coefficients in first row of w. And the projections for obtaining other coefficients in w are performed state variable by state variable.

Note the basis functions chosen in the test problem act as a way of providing good enough approximation to the vector field. Other non-linear basis functions such as set of higher degree monomials could also be further investigated as a way of approximating complex vector field with higher accuracy.

As mentioned above, the Koopman-based technique could be sensitive to the observation noise. The denoising method forward/backward DMD (fbDMD) proposed in [9] is adopted for the numerical experiments reported below. Forward/backward DMD computes the matrices approximating discrete evolution of the dynamical system forward in time as well as for the dynamics backward in time, by swapping M_1 and M_2 (i.e., in forward-time each observed state variable snapshot pair transits from t_i to t_{i+1} while in backward-time from t_{i+1} back to t_i). The matrix U^b approximating the linear operator of the reverse dynamical system should be expected to be an estimate of the inverse of the matrix U obtained in the forward pass, with $\mathrm{U} \approx (\mathrm{U}^b)^{-1}$. Once approximations for the forward-time and backward-time matrices are obtained, a denoised estimate of the discrete evolution in (3.9) can be determined by combining U and U^b in the following way,

$$\widetilde{\mathrm{U}} = \left(\mathrm{U}(\mathrm{U}^b)^{-1}\right)^{\frac{1}{2}} \tag{3.13}$$

To solve $\widetilde{\mathrm{U}}$ in (3.13), either a matrix square root finder or matrix logarithms could be considered. Since matrix logarithms are required to obtain an approximation for the vector field, we adopt the proposed alternative in [9] by taking the average of the matrix logarithms, with $\mathrm{L} = \frac{1}{\Delta t}\log \mathrm{U}$ and $\mathrm{L}^b = \frac{1}{\Delta t}\log(\mathrm{U}^b)^{-1}$,

$$\widetilde{\mathrm{L}} = \frac{1}{2}\left(\mathrm{L} + \mathrm{L}^b\right) \tag{3.14}$$

Then a denoised approximation to the vector field can be obtained and vector field values are calculated in the subsequent steps with $\widetilde{\mathrm{L}}$ to identify the coefficients w.

Unscented Kalman Filtering (UKF)

Kalman Filtering (KF) [18] was first introduced for state variable estimation and correction from noisy measurements under (known) linear dynamics. Extended Kalman Filtering (EKF) and Unscented Kalman Filtering (UKF) [17] were later proposed to deal with nonlinear dynamics during the process. EKF linearizes the nonlinear dynamics by considering multi-dimensional Taylor expansions to obtain the model-predicted next states. UKF applies the nonlinear dynamics on a set of carefully selected deterministic points (called sigma points). These points provide approximation of the moments of the filter density up to a certain order, which is in the class of statistical linearization schemes [11]. Specifically, assume that $\mathrm{Y}_i \equiv y(t_i) \in \mathbb{R}^{n_y}$ and $\hat{\mathrm{Y}}_i \equiv \hat{y}(t_i) \in \mathbb{R}^m$ are the state and observation vectors at time t_i. The discrete time difference equation is then,

$$\mathrm{Y}_i = \mathrm{T}(\mathrm{Y}_{i-1}), \tag{3.15}$$

$$\hat{\mathrm{Y}}_i = \mathrm{V}(\mathrm{Y}_i) + \eta_i, \tag{3.16}$$

where $\mathrm{T}(\mathrm{Y}_{i-1})) = \mathrm{Y}_{i-1} + \int_{t_{i-1}}^{t_i} f(s, y(s), p)ds$ with f being the right-hand-side function in (1.1), and η_i is the uncorrelated Gaussian observation noise at time t_i. V is the observation function and in our experiments, V is simply the identity and $m = n_y$. Note that additional process noise can be considered in (3.15), but in the current investigation, only deterministic dynamics is studied.

The Kalman filter approach consists of a prediction and a correction step. Given the previous observations $\hat{\mathrm{Y}}_{1:i} = \{\hat{\mathrm{Y}}_1, \hat{\mathrm{Y}}_2, \ldots, \hat{\mathrm{Y}}_i\}$ up to time t_i and the evolution dynamics T, Kalman filter predicts the most likely state estimation $\mathrm{z}(t_i|t_{i-1})$ and its associated observations $\hat{\mathrm{z}}(t_i|t_{i-1})$,

$$\mathrm{z}(t_i|t_{i-1}) = \mathbb{E}[\mathrm{Y}_i|\hat{\mathrm{Y}}_{1:i}] = \mathbb{E}[\mathrm{T}(\mathrm{Y}_{i-1})|\hat{\mathrm{Y}}_{1:i}], \tag{3.17}$$

$$\hat{\mathrm{z}}(t_i|t_{i-1}) = \mathbb{E}[\hat{\mathrm{Y}}_i|\hat{\mathrm{Y}}_{1:i}] = \mathbb{E}[\mathrm{V}(\mathrm{Y}_i)|\hat{\mathrm{Y}}_{1:i}], \tag{3.18}$$

And the associated covariances or prediction error in the estimation is given by:

$$\mathrm{P}(t_i|t_{i-1}) = \mathbb{E}[(\mathrm{Y}_i - \mathrm{z}(t_i|t_{i-1}))(\mathrm{Y}_i - \mathrm{z}(t_i|t_{i-1}))^T|\hat{\mathrm{Y}}_{1:i}], \tag{3.19}$$

$$\mathrm{P}_{\hat{\mathrm{Y}}\hat{\mathrm{Y}}}(t_i|t_{i-1}) = \mathbb{E}[(\hat{\mathrm{Y}}_i - \hat{\mathrm{z}}(t_i|t_{i-1}))(\hat{\mathrm{Y}}_i - \hat{\mathrm{z}}(t_i|t_{i-1}))^T|\hat{\mathrm{Y}}_{1:i}], \tag{3.20}$$

$$\mathrm{P}_{\mathrm{Y}\hat{\mathrm{Y}}}(t_i|t_{i-1}) = \mathbb{E}[(\mathrm{Y}_i - \mathrm{z}(t_i|t_{i-1}))(\hat{\mathrm{Y}}_i - \hat{\mathrm{z}}(t_i|t_{i-1}))^T|\hat{\mathrm{Y}}_{1:i}]. \tag{3.21}$$

Then in the correction step, Kalman filter updates the predictions for the state and estimation error using the new observation $\hat{y}(t_i)$:

$$\mathrm{z}(t_i|t_i) = \mathrm{z}(t_i|t_{i-1}) + \mathrm{K}_i(\hat{y}(t_i) - \hat{\mathrm{z}}(t_i|t_{i-1})), \tag{3.22}$$

$$\mathrm{P}(t_i|t_i) = \mathrm{P}(t_i|t_{i-1}) - \mathrm{K}_i\mathrm{P}_{\hat{\mathrm{Y}}\hat{\mathrm{Y}}}(t_i|t_{i-1})\mathrm{K}_i^T, \tag{3.23}$$

where K_i is the *Kalman gain matrix* at time t_i that represents the uncertainties given by the errors of the previous prediction step and it is defined as

$$\mathrm{K}_i = \mathrm{P}_{\mathrm{Y}\hat{\mathrm{Y}}}(t_i|t_{i-1})\mathrm{P}^{-1}_{\hat{\mathrm{Y}}\hat{\mathrm{Y}}}(t_i|t_{i-1}). \tag{3.24}$$

For linear dynamics, equations (3.17) and (3.18) can be solved analytically. For nonlinear dynamics, the difference between EKF and UKF comes from the different treatment of equations (3.17) and (3.18). EKF deals with the nonlinear dynamics by local linearization and UKF applies the nonlinear dynamical transformation (i.e., performing the integration) to the set of sigma points $\{\mathcal{X}_l\}_{l=0}^{2n_y}$ selected by

$$\begin{aligned}
\mathcal{X}_0(t_{i-1}|t_{i-1}) &= \mathrm{z}(t_{i-1}|t_{i-1}),\\
\mathcal{X}_l(t_{i-1}|t_{i-1}) &= \mathrm{z}(t_{i-1}|t_{i-1}) + \sqrt{(n_y+\kappa)\mathrm{P}(t_{i-1}|t_{i-1})},\\
\mathcal{X}_{l+n_y}(t_{i-1}|t_{i-1}) &= \mathrm{z}(t_{i-1}|t_{i-1}) - \sqrt{(n_y+\kappa)\mathrm{P}(t_{i-1}|t_{i-1})},
\end{aligned} \tag{3.25}$$

with $l = 1, \ldots, n_y$. The value of κ is tunable but was fixed to zero in the experiments reported here for simplicity.

Nonlinear evolution and its associated observations are applied to the set of sigma points:

$$\mathcal{X}_l(t_i|t_{i-1}) = \mathrm{T}(\mathcal{X}_l(t_{i-1}|t_{i-1})), \tag{3.26}$$

$$\mathcal{Y}_l(t_i|t_{i-1}) = \mathrm{V}(\mathcal{X}_l(t_i|t_{i-1})), \tag{3.27}$$

and equations (3.17) and (3.18) can be approximated by

$$\mathrm{z}(t_i|t_{i-1}) = \sum_{l=0}^{2n_y} W_l \mathcal{X}_l(t_i|t_{i-1}), \tag{3.28}$$

$$\hat{\mathrm{z}}(t_i|t_{i-1}) = \sum_{l=0}^{2n_y} W_l \mathcal{Y}_l(t_i|t_{i-1}), \tag{3.29}$$

and covariance

$$\mathrm{P}_{\hat{\mathrm{Y}}\hat{\mathrm{Y}}}(t_i|t_{i-1}) = \sum_{l=0}^{2n_y} W_l \{\mathcal{Y}_l(t_i|t_{i-1}) - \hat{\mathrm{z}}(t_i|t_{i-1})\}\{\mathcal{Y}_l(t_i|t_{i-1}) - \hat{\mathrm{z}}(t_i|t_{i-1})\}^T, \tag{3.30}$$

$$\mathrm{P}_{\mathrm{Y}\hat{\mathrm{Y}}}(t_i|t_{i-1}) = \sum_{l=0}^{2n_y} W_l \{\mathcal{X}_l(t_i|t_{i-1}) - \mathrm{z}(t_i|t_{i-1})\}\{\mathcal{Y}_l(t_i|t_{i-1}) - \hat{\mathrm{z}}(t_i|t_{i-1})\}^T, \tag{3.31}$$

$$\mathrm{P}(t_i|t_{i-1}) = \sum_{l=0}^{2n_y} W_l \{\mathcal{X}_l(t_i|t_{i-1}) - \mathrm{z}(t_i|t_{i-1})\}\{\mathcal{X}_l(t_i|t_{i-1}) - \mathrm{z}(t_i|t_{i-1})\}^T, \tag{3.32}$$

where the weights $W_l, l = 0, \ldots, 2n_y$ are defined as,

$$W_0 = \frac{\kappa}{n_y + \kappa}, \tag{3.33}$$

$$W_l = \frac{1}{2(n_y + \kappa)}, (l = 1, \ldots, 2n_y). \tag{3.34}$$

In a subsequent investigation of [31], UKF was used in parameter estimation by augmenting the parameters as additional state variables with zero dynamical evolution. It starts with observation point at the initial time, with both the original state variables and the augmented parameters updated as in UKF, and the filtering process proceeds until the end of the trajectory.

For example in the Calcium Ion problem, the transition dynamics T on the state variables with the linear parameters as the augmented states can be written as,

$$\mathrm{T}(\mathrm{Y}_{i-1}) = \mathrm{T}(\begin{bmatrix} y_1(t_{i-1}) \\ y_2(t_{i-1}) \\ y_3(t_{i-1}) \\ y_4(t_{i-1}) \\ k_1 \\ \vdots \\ k_{11} \end{bmatrix}) = \begin{bmatrix} y_1(t_{i-1}) \\ y_2(t_{i-1}) \\ y_3(t_{i-1}) \\ y_4(t_{i-1}) \\ k_1 \\ \vdots \\ k_{11} \end{bmatrix} + \begin{bmatrix} \int_{t_{i-1}}^{t_i} f_1(s, y(s), p)ds \\ \int_{t_{i-1}}^{t_i} f_2(s, y(s), p)ds \\ \int_{t_{i-1}}^{t_i} f_3(s, y(s), p)ds \\ \int_{t_{i-1}}^{t_i} f_4(s, y(s), p)ds \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \tag{3.35}$$

where the last 11 components of Y_{i-1} are the linear parameters (augmented states) with zero dynamics (i.e., parameters in the system of ODEs (2.1)-(2.4) are assumed to be constant throughout the interval of interest). Once the transition dynamics T is specified, other steps in the UKF approach can be followed to provide estimation of the predicted state variables including the parameters, and the uncertainty of the estimation.

A natural way to combine UKF with INT-SME is to use UKF to refine the crude initial guess obtained by INT-SME. However, this approach trades a higher cost for a nonlinear dynamics simulation in every timestep during the filtering to obtain a more accurate initial guess for the second stage.

Gaussian Process (GP)

A Gaussian process [36] can be viewed as a distribution over functions, while its inference takes place directly in the function space. It is a collection of random variables, any finite number of which have a joint Gaussian distribution. It is a non-parametric probabilistic model for function estimation that is widely used in tasks such as regression and classification, and in our case, the random variables are the function values at specific locations. A Gaussian process is completely specified by its mean function $\mu(t)$ and covariance function or *kernel*: $k(t, t')$. For a multi-dimensional state variable, a Gaussian process regression could be used as smoother for each state variable with function values at points outside of recorded time points inferred in the following way. For a specific state variable y_j, with zero mean and squared exponential covariance prior ($k(t, t') = \exp(-\frac{1}{2}|t - t'|^2)$) and all the available training input points (stacked recorded time points) $D = [t_1, \ldots, t_{n_o}]^T \in \mathbb{R}^{n_o}$, to infer function values on new test points (time points outside of recorded timestamps) $D_* = [t_1^*, \ldots, t_{n_t}^*]^T \in \mathbb{R}^{n_t}$, function values at training points $\mathrm{s} = [y_j(t_1), \ldots, y_j(t_{n_o})]^T$ as well as the approximated function values at test points $\mathrm{s}_* = [\tilde{y}(t_1^*), \ldots, \tilde{y}(t_K^*)]^T$ satisfy the prior distribution:

$$\begin{bmatrix} \mathrm{s} \\ \mathrm{s}_* \end{bmatrix} \sim \mathcal{N}(0, \begin{bmatrix} K(D,D) & K(D,D_*) \\ K(D_*,D) & K(D_*,D_*) \end{bmatrix}), \tag{3.36}$$

where $K(D, D_*)$ denotes the matrix with each entry being the covariances evaluated at all pairs of training and test points, and similarly for the other matrices $K(D, D)$, $K(D_*, D_*)$ and $K(D_*, D)$. The posterior distribution of s_* is then given by

$$\begin{aligned} \mathrm{s}_*|D_*, D, \mathrm{s} \sim \mathcal{N}(&K(D_*,D)K(D,D)^{-1}\mathrm{s}, \\ &K(D_*,D_*) - K(D_*,D)K(D,D)^{-1}K(D,D_*)). \end{aligned} \tag{3.37}$$

When considering noisy observations $\hat{y}(t) = y(t) + \epsilon$ where $\epsilon \sim \mathcal{N}(0, \sigma^2)$, distribution (3.36) is replaced by

$$\begin{bmatrix} \hat{\mathrm{s}} \\ \mathrm{s}_* \end{bmatrix} \sim \mathcal{N}(0, \begin{bmatrix} K(D,D) + \sigma^2 I & K(D,D_*) \\ K(D_*,D) & K(D_*,D_*) \end{bmatrix}), \tag{3.38}$$

where $\hat{\mathrm{s}} = [\hat{y}_j(t_1), \ldots, \hat{y}_j(t_{n_o})]^T$ and the posterior distribution is derived accordingly. The mean function of the posterior distribution can be used as a smoother for the observed data.

The kernel of choice reflects the covariance assumptions for the underlying process to be modeled. There have been several investigations [4, 35, 34] of applying GP to parameter estimation that were largely inspired by the SME approach to parameter estimation by treating the Gaussian process approximated curve as a smoother for the observed trajectory.

In our experiments, Gaussian process regression is used to obtain smoothed curves for each state variable. The standard MATLAB Gaussian Regression Regression routine `fitrgp` is used as a way for smoothing the observed trajectories. Take the Calcium Ion problem as an example, regression values for the state variables are sought at their original observed time points, i.e., t_i^* is set to be t_i. Focus on the first state variable for example, and stack the observed state variables over the interval of interest as $\mathrm{s} = [\hat{y}_1(t_1), \ldots, \hat{y}_1(t_{n_o})]^T$. A Gaussian process regression is then conducted by first populating the matrix $K(D, D)$ (and similarly $K(D_*, D)$) as,

$$\begin{bmatrix} \exp(-\frac{1}{2}|t_1 - t_1|^2) & \exp(-\frac{1}{2}|t_1 - t_2|^2) & \dots & \exp(-\frac{1}{2}|t_1 - t_{n_o}|^2) \\ \exp(-\frac{1}{2}|t_2 - t_1|^2) & \exp(-\frac{1}{2}|t_2 - t_2|^2) & \dots & \exp(-\frac{1}{2}|t_2 - t_{n_o}|^2) \\ \vdots & \vdots & \ddots & \vdots \\ \exp(-\frac{1}{2}|t_{n_o} - t_1|^2) & \exp(-\frac{1}{2}|t_{n_o} - t_2|^2) & \dots & \exp(-\frac{1}{2}|t_{n_o} - t_{n_o}|^2) \end{bmatrix} \tag{3.39}$$

with the simple squared exponential kernel for illustrating purpose. And the regression values s_* of the state variable y_1 at the observed times points can then be obtained by performing inference over the probability distribution (3.38).

Once the regression values s_* is obtained for the first state variable y_1, similar regression process can be performed for the other state variables. Note that in the Gaussian process approach, no model information from the system of ODEs is incorporated into the regression model. Further investigation could be left as future work to explore a more informed inference process combining the specific parameter estimation task for a system of ODEs with the Gaussian process regression.

3.1.1 Techniques to include a wider range of initial guesses

Even with the two-stage procedure and a cheaper objective function to evaluate so that suitable parameters can be obtained in the first stage, there is still a chance of only finding suboptimal local minimum as can be observed in the results for the Calcium Ion problem in Chapter (4). One could also consider running the first stage optimization over a subset of observations, or even statistically generated trajectories to provide more candidate initial guesses to consider before starting the second stage.

Progressive Shooting (PS)

Integration of the IVP (1.1) with parameter values far from the true ones often leads to a numerical solution that only remains close to observation points near the initial time. Based on this observation, Incremental Shooting [24] and Progressive Shooting [20] have been motivated and introduced to circumvent the potential risk of failure of the simulation with certain parameter values before the final time T or the discovery of a poor local minimum that would otherwise occur if further fitting with those inappropriate parameter values were continued. In implementation, Progressive shooting proceeds by optimizing parameters over a subset of the observation points near the initial observation, and progressively include increasing observation points over a longer interval in the definition of the objective function until the end of the interval of interest is included.

Calver et al. [6] proposed the use of INT-SME combined with the ideas from PS (named, PS-INT-SME) and the proposed technique effectively improved the performance on problem such as the linear parameter estimation in the Calcium Ion problem.

Datapoint Perturbation

One of the major benefits of PS-INT-SME is that they can provide multiple promising candidate starting vector to consider for the second stage optimization. These initial guesses are obtained by having different but related objective functions that depend on different parts of the trajectory. A different approach for providing additional initial trials could be to perturb the datapoints, either the smoothed/estimated trajectory or the originally collected observations, according to the probability distribution of the noise assumption or the probability model of the smoothed trajectory. The smoothed trajectory acts as an approximation to the underlying true trajectory, and generating additional trajectory samples is a process of obtaining additional sets of similar observations. These generated trajectory samples could be used to form different objective functions, thus giving rise to additional initial trials for better chance of escaping suboptimal local minimum if considering only the original noisy observations. A collection of initial guesses, including the one obtained with the original observations, will be ranked according to the objective function involving full simulation, to run in the second stage.

In terms of how the previous techniques can be used with datapoint perturbation, some of the models already enjoy a statistical derivation and could be directly suitable for trajectory sample generating, for example Gaussian process. However in the following experiments, noise with variance relative to the magnitude of the estimated trajectory is added to generate additional trajectories. This is consistent with the noise assumption in the test problems. To use the model itself to generate trajectory samples, additional effort should be made to derive a variation of the model to reflect the relative noise setting (which is left as future investigation). For UKF, the filtering process already considers the influence of noise and produces an estimated mean and covariance of the parameters. We will use this mean and covariance estimate to generate additional parameters to include in the collection of initial guesses.

For PINN, Koopman and INT-SME, there is no probabilistic justification for these techniques and they are derived from a deterministic point of view. Instead, noise relative to the magnitude of the observed trajectory (for Koopman and INT-SME) or the smoothed trajectory (for PINN) is added to generate additional trajectories and obtain additional promising trial parameters to consider for the initial guesses.

3.1.2 Integration of the Above Techniques

In summary, the different techniques introduced above can be integrated into the two stage procedure using INT-SME in the following ways,

- PINNs and GPs are used to smooth the data then run INT-SME on the smoothed curve and/or the original observed trajectory. Note that PINNs could also provide initial guesses for the parameters if the technique for identifying the underlying differential equation parameters in [27] is used.
- UKF could provide both a filtered trajectory and a refined initial guess of p_0.
- Koopman technique provides its own initial guess by estimating the vector field with a different theoretical background, which is an independent way of generating the initial guess for the parameters.

The second stage optimization with the objective function (1.2) evaluated by full simulation of the IVP will be run after each of the above techniques.

When datapoint perturbation is used in combination with the above methods,

- For PINNs and GP, artificial relative noise is added to PINN- and GP-smoothed curve to generate additional trajectories to obtain more promising candidates for the initial guesses by INT-SME.
- For the Koopman technique, the observed trajectory is further perturbed by noise relative to the magnitude of the observed state variables, before populating the matrices.
- For UKF, additional initial guesses are drawn from the resulting probability distribution that describes the uncertainty of the filtered parameters.

In all of the above cases, a total of N_P sets of perturbed data points/additional initial guesses are drawn from each individual distribution and we found that $N_P = 20$ would give reasonable results. Note that to encourage a larger number of promising parameter candidates to be included in the trials, the initial guess (or the refined initial guess) by each method running on the original observed data will be kept.

3.2 Total Least Squares

It is well-known that if the independent variables contain errors, then the results of linear regression produce biased estimates. When formulating the linear least square matrix from equation (3.4), if all parameters appear linearly in (1.1) and $\overline{G} = 0$, the least square problem becomes,

$$\min_r \int_0^T \Big\| \big(\tilde{y}(t) - y_0\big) - \big[\int_0^t G(s, \tilde{y}(s))\, ds\big] r \Big\|^2 \, dt, \tag{3.40}$$

where r is the vector of linear parameters. Denote the matrix representing the integration in (3.40) as,

$$\int_0^t G(s, \tilde{y}(s))\, ds \equiv \tilde{A}(t), \tag{3.41}$$

and denote $\tilde{y}(t) - y_0 \equiv \tilde{b}(t)$. It should be noted that the ordinary least square approach omits the fact that independent variables in (3.40) also contain errors, i.e., $\tilde{y}(t)$ obtained from the noisy observations appear in both $\tilde{A}(t)$ and $\tilde{b}(t)$, not just $\tilde{b}(t)$ as in ordinary least square regression. This suggests we should consider using an error-in-variables approach, where errors in both $\tilde{A}(t)$ and $\tilde{b}(t)$ of (3.40) are considered and addressed with some form of total least square method. In addition, the formulation (3.41) may lead to the fact that, noise originally appearing in the standalone state variables may also propagate nonlinearly in the matrix $\tilde{A}(t)$ through G and results in certain correlation between the entries in $\tilde{A}(t)$ as well as with components of $\tilde{b}(t)$.

We start by considering a simple example, the Barnes problem, which demonstrates the potential difficulties. In this problem, $\tilde{A}(t)$ can be written as,

$$\int_0^t \begin{bmatrix} \tilde{y}_1(s) & -\tilde{y}_1(s)\tilde{y}_2(s) & 0 \\ 0 & \tilde{y}_1(s)\tilde{y}_2(s) & -\tilde{y}_2(s) \end{bmatrix} ds \tag{3.42}$$

The errors and correlations between $y_1(t)$, $y_2(t)$, $\int_0^t y_1(s)ds$, $\int_0^t y_2(s)ds$, as well as $\int_0^t y_1(s)y_2(s)ds$ could be intricate. If we choose a smoother to interpolate exactly the observed state variable points, with $\tilde{y}(t_i) = \hat{y}(t_i)$, then, since $y_1(t)$ and $y_2(t)$ are assumed to be normally distributed, it follows that their integrals exhibit Brownian motion, with variance, t (scaled by the noise in the observations, σ). For the other integral, the product of two normally distributed random variables is a sum of Chi-squared random variables. Then the integral should also be Chi-squared.

For simplicity without loss of generality, sum over the integrand of the outer integration in (3.40) is used rather than a quadrature rule, i.e., $\int_0^T \|\tilde{b}(t) - \tilde{A}(t)r\|^2 \, dt \approx \sum_{i=1}^{n_o} \|\tilde{b}(t_i) - \tilde{A}(t_i)r\|^2 \Delta t$. Denote $\tilde{A}_j = [\tilde{A}_j(t_1); \tilde{A}_j(t_2); \ldots; \tilde{A}_j(t_{n_o})]$, for $j = 1, \ldots, n_y$

where $\tilde{A}_j(t)$ is the jth row of matrix $\tilde{A}(t)$. Similarly, let $\tilde{b}_j = [\tilde{b}_j(t_1); \tilde{b}_j(t_2); \ldots; \tilde{b}_j(t_{n_o})]$. Then by stacking $\tilde{A}_j$'s and $\tilde{b}_j$'s, the least-square problem to be solved can be expressed in the form of minimizing $\|\mathrm{A}r - \mathrm{b}\|^2$, where the left-hand-side matrix A and right-hand-side vector b become,

$$\mathrm{A} \equiv \begin{bmatrix} \tilde{A}_1 \\ \tilde{A}_2 \\ \vdots \\ \tilde{A}_{n_y} \end{bmatrix} \in \mathbb{R}^{n_y \cdot n_y \times n_p}, \mathrm{b} \equiv \begin{bmatrix} \tilde{b}_1 \\ \tilde{b}_2 \\ \vdots \\ \tilde{b}_{n_y} \end{bmatrix} \tag{3.43}$$

Note for the SME approach, matrix A can be obtained following a very similar process by stacking $G_i = [G_i(t_1); G_i(t_2); \ldots; G_i(t_{n_o})]$ and the corresponding time derivative approximation by the smoother. Also, if the parameter estimation problem can be decomposed into several independent subproblems, the matrix A and vector b can be constructed by including only relevant terms in each subproblem and the correpsonding least-square problems can be solved independently. Dimensions of A and b are adjusted according to the number of linear parameters and number of state variables involved in each subproblem respectively.

There have been investigations [12] in total least squares method for error-in-variables models in which perturbation are allowed in both the left-hand-side data matrix and right-hand-side observation vector. The approach proposed in [12] uses a Singular Value Decomposition (SVD) to obtain a minimizer of the following objective function in the Frobenius norm,

$$\begin{aligned} &\min_{\delta\mathrm{A},\delta\mathrm{b}} \|[\delta\mathrm{A}, \delta\mathrm{b}]\|_F^2, \\ &\text{s.t: } \mathrm{b} + \delta\mathrm{b} \in \mathit{Range}(\mathrm{A} + \delta\mathrm{A}). \end{aligned} \tag{3.44}$$

Let $[\mathrm{A}, \mathrm{b}] = \sum_{i=1}^{n_p+1} \sigma_i u_i v_i^T$ be the SVD of $[\mathrm{A}, \mathrm{b}]$. Then the corresponding total least square solution for r can be defined in terms of the associated eigenvalue problem,

$$\begin{bmatrix} \mathrm{A}^T\mathrm{A} & \mathrm{A}^T\mathrm{b} \\ \mathrm{b}^T\mathrm{A} & \mathrm{b}^T\mathrm{b} \end{bmatrix} \cdot \begin{bmatrix} r \\ -1 \end{bmatrix} = \sigma_{n_p+1}^2 \begin{bmatrix} r \\ -1 \end{bmatrix} \tag{3.45}$$

and the TLS minimizer r_{TLS}^* is given by,

$$r^*_{TLS} = (\mathrm{A}^T\mathrm{A} - \sigma^2_{n_p+1} I)^{-1}\mathrm{A}^T\mathrm{b}. \tag{3.46}$$

We present the experimental results in Section (4.2). Note that for the standard TLS method in [12], diagonal weighting matrices are also allowed. Although improvement can be observed in several test problems with the standard TLS technique, we also notice that the optimization setting in the standard total least square might still not be adequate. $\|[\delta\mathrm{A}, \delta\mathrm{b}]\|^2_F$ might be minimized in a uniformly weighted way. If there is complicated correlation between the entries in the left-hand-side matrix A or even correlation between A and b, it may prove useful and beneficial to consider a weighted total least square approach. We have noticed a weighted total least-squares (WTLS) solution [30] has been proposed to deal with the problems in which the observations can be heteroscedastic and correlated, and we leave the investigation of this alternative approach to future work.

Chapter 4

Experimental Results

To test the aforementioned techniques, we consider a set of test problems from the ODE parameter estimation literature (see Section (2)). In these test problems, we seek to deduce a parameterized model using systems of ODEs with observed data from experiments. The parameters for the models are estimated by optimizing the objective function (1.2) that is obtained by comparing the simulated trajectory against the observed state variables. The true trajectory is obtained using a reliable numerical solver such as DDEM [39] to integrate (1.1) with true parameters. 100 sets of observed data are then generated by adding relative noise to the exact trajectory for each test problem.

All of the above techniques will be used to generate initial guesses, and a gradient-based optimizer will be used to minimize (1.2) in the second stage. Model sensitivities are approximated by finite difference as provided in MATLAB `lsqnonlin` routine with default settings. We use the UKF implementation in [7] in our experiments involving UKF.

4.1 Experiment Settings and Results

The following section explains several experimental considerations and/or hyperparameters used for the numerical experiments.

In our experiments using the Koopman-based lifting techniques, the Koopman basis functions chosen for the Calcium Ion problem are monomials up to the highest degree of terms appearing in the right-hand-side of equations (2.1)-(2.4), plus other non-polynomial expressions in these equations (see for example the populated matrix (3.12) to be solved in Section (3), and the nonlinear parameters are treated as known). For the Barnes problem, monomials of degree up to 2 are used as basis functions, since 2 is the highest polynomial degree in the right-hand-side of (2.5) and (2.6). For the Mendes problem, monomials up to degree 1, and 9 non-polynomial expressions that appear in the right-hand-side of equations (2.10)-(2.17) are chosen as basis functions, e.g., $\frac{y_1}{y_1+q_{13}}$ in (2.13). For the Goodwin problem, monomials of degree up to 1 and one non-polynomial term (i.e., $\frac{1}{A+y_3(t)^\sigma}$ in (2.7)) are chosen as basis functions. In addition to the choice of basis functions, we also note that there can be an issue that arises when attempting to avoid complex-valued matrices when calculating the matrix logarithm L. In the current implementation, the real part of L is kept to provide initial guesses of the parameters for further optimization in the second stage.

For PINNs technique, a neural network with 8 hidden layers is implemented to provide smoothed trajectories for each state variable in each of the test problems, and the numbers of hidden units in each hidden layer are given by 20, 20, 40, 80, 80, 40, 20, 20.

For GP, the kernel of choice is determined from experimental assessment, and we note that for Barnes problem, the Matérn kernel with parameter $\frac{3}{2}$ (see [36] for more detail) would yield reasonable results. For the Calcium Ion problem, the Goodwin problem and the Mendes problem, we choose the squared exponential kernel.

For UKF, initial guess of the parameters provided by INT-SME along with the (given) initial conditions will be used as the starting values of the augmented state variables at time zero and filtered until the end of the time interval for one pass. For the Mendes problem, we find that filtering the trajectory corresponding to the pair ($P = 0.05$ and $S = 0.1$) gives reasonable results that takes into the consideration the trade-off between obtaining refined parameter and running UKF without spending a large amount of effort in filtering using some of the other trajectories.

In the previous sections and the experiments, initial conditions in the test problems are assumed to be known and simulation is run with those given initial conditions used for the evaluation of objective function (1.2). If the initial conditions are assumed to be unknown, they can be included as additional parameters to be estimated. With the INT-SME technique, the initial conditions appear linearly in the objective function (3.4), and the initial guesses for them can be obtained in one iteration of the associated linear least square problem together with the other linear parameters. This does not make much difference in the effort required in the first stage. For the second stage, we also observed very little difference between whether initial conditions are unknown or known, even if additional estimation for the initial conditions has to be performed. The experimental results are included in Tables (4.1)-(4.5) for each technique considered running on the Calcium Ion problem, Mendes problem, Barnes problem, and Goodwin problem respectively. The observation points are uniformly spaced along the interval of interest.

	INT-SME					
Test Problems	INT-SME only		perturbed datapoints		progressive shooting	
	success	avg iters	success	avg iters	success	avg iters
Calcium Ion	73/100	41.4247	98/100	45.4082	100/100	22.0200
Mendes	100/100	5.8400	100/100	5.8400	100/100	5.6800
Barnes	100/100	4.3800	100/100	3.3600	100/100	3.2500
Goodwin	100/100	7.2500	100/100	5.7600	100/100	5.8900

Table 4.1: Results of running parameter estimation on 100 noisy datasets using INT-SME, with the nonlinear parameters fixed, on uniform grids. avg iters are the average number of iterations in stage II among success runs.

	PINN					
Test Problems	PINN-smoothed only		perturbed datapoints		progressive shooting	
	success	avg iters	success	avg iters	success	avg iters
Calcium Ion	31/100	88.1290	90/100	118.0000	78/100	273.5256
Mendes	100/100	5.9700	100/100	5.9700	100/100	5.6900
Barnes	100/100	4.3600	100/100	3.3200	100/100	3.2500
Goodwin	100/100	6.6100	100/100	6.0100	100/100	5.9400

Table 4.2: Results of running parameter estimation on 100 noisy datasets using PINN, with the nonlinear parameters fixed, on uniform grids.

	Koopman					
Test Problems	Koopman only		perturbed datapoints		progressive shooting	
	success	avg iters	success	avg iters	success	avg iters
Calcium Ion	89/100	34.9775	98/100	63.0714	97/100	23.2784
Mendes	96/100	8.2396	100/100	6.8400	100/100	6.0600
Barnes	100/100	4.1200	100/100	3.3100	100/100	3.6800
Goodwin	91/100	16.3297	99/100	13.3535	96/100	12.3021

Table 4.3: Results of running parameter estimation on 100 noisy datasets using Koopman, with the nonlinear parameters fixed, on uniform grids.

	GP					
Test Problems	GP-smoothed only		perturbed datapoints		progressive shooting	
	success	avg iters	success	avg iters	success	avg iters
Calcium Ion	45/100	45.8000	99/100	101.4949	99/100	25.8182
Mendes	100/100	6.1600	100/100	6.1600	100/100	5.0800
Barnes	100/100	4.4500	100/100	3.3700	100/100	3.2500
Goodwin	100/100	6.6800	100/100	6.1400	100/100	5.7700

Table 4.4: Results of running parameter estimation on 100 noisy datasets using GP, with the nonlinear parameters fixed, on uniform grids.

Test Problems	UKF					
	UKF-refined p_0 only		perturbed p_0		progressive shooting	
	success	avg iters	success	avg iters	success	avg iters
Calcium Ion	87/100	21.6437	94/100	33.4574	100/100	10.8600
Mendes	100/100	9.3500	100/100	8.6700	99/100	8.7778
Barnes	100/100	3.2700	100/100	2.8800	100/100	3.0600
Goodwin	100/100	5.4900	100/100	4.9100	100/100	5.7100

Table 4.5: Results of running parameter estimation on 100 noisy datasets using UKF, with the nonlinear parameters fixed, on uniform grids. Note perturbed p_0's are generated by sampling from the estimated mean and covariance of the filtered p_0 in UKF.

4.2 Numerical Experiments on TLS

To demonstrate the effectiveness of considering an Error-in-Variable model and applying the TLS approach, we present the numerical experiments conducted on the Barnes problem, Calcium Ion problem, Goodwin problem and Mendes problem. See Table (4.6) for the results of using the INT-SME formulation and Table (4.7) using the SME formulation.

Test Problems	INT-SME		INT-SME TLS	
	success	avg iters	success	avg iters
Barnes	100/100	4.3800	100/100	4.2200
Calcium Ion	76/100	42.6053	100/100	21.0600
Goodwin	100/100	6.5900	100/100	5.9400
Mendes	100/100	5.8400	100/100	5.8000

Table 4.6: TLS results for various test problems, with the nonlinear parameters fixed and using INT-SME compared with their ordinary least squares counterparts. The table shows the result of running parameter estimation on 100 noisy datasets. avg iters are the average number of iterations in stage II among success runs.

Test Problems	SME		SME TLS	
	success	avg iters	success	avg iters
Barnes	100/100	5.7000	100/100	5.3000
Calcium Ion	96/100	44.8750	95/100	26.4526
Goodwin	89/100	24.6742	100/100	9.3200
Mendes	100/100	7.5100	100/100	7.1300

Table 4.7: TLS results for various test problems, with the nonlinear parameters fixed and using SME compared with their ordinary least squares counterparts. The table shows the result of running parameter estimation on 100 noisy datasets. avg iters are the average number of iterations in stage II among success runs.

In each test problem, the TLS approach could either achieve higher success rate, or improve the initial guess so that less effort is necessary in the second stage (in terms of average number of iterations) before converging to an acceptable minimizer. The improvement is especially obvious when INT-SME is applied to the Calcium Ion problem, and SME is applied to the Goodwin problem, leading to both a significant increase in the success rate and a more suitable parameter initial guess to start the second stage.

4.3 Experiments Allowing Further Query of Data

When the trajectory in the test problem contains regions in which the state variables change rapidly, a denser distribution of data points may be needed for smoother approximations in those regions. To account for the potential smoothness issue in interpolation, additional data points are allowed to be queried on regions detected by the techniques where the state variables change rapidly. The regions are detected by considering integration errors of the integral (3.4), which is asymptotically proportional to the difference between trapezoidal rule approximation with finer and coarser observation grids. Observation points at regions with lower estimated error will be traded for denser observation points in regions with higher estimated error to keep the overall number of observation points constant. See Figure (4.1) for a sample plot of the queried mesh points, true trajectories and observed points. The experimental results on the resulting non-uniform grids are included in Tables (4.8)-(4.11) for each techniques considered running on the Calcium Ion problem, Mendes problem, Barnes problem, and Goodwin problem respectively, except Koopman, for which further investigation is required before the approach can be extended to a non-uniform grid.

	INT-SME					
Test Problems	INT-SME only		perturbed datapoints		progressive shooting	
	success	avg iters	success	avg iters	success	avg iters
Calcium Ion	84/100	48.1310	100/100	60.1800	100/100	18.6100
Mendes	100/100	5.7700	100/100	5.7700	100/100	5.7700
Barnes	100/100	4.1500	100/100	3.3700	100/100	3.2600
Goodwin	100/100	6.4100	100/100	5.7400	100/100	5.5600

Table 4.8: Results of running parameter estimation on 100 noisy datasets using INT-SME, with the nonlinear parameters fixed, on non-uniform grids that allow further query of data on rapidly changing regions. avg iters are the average number of iterations in stage II among success runs.

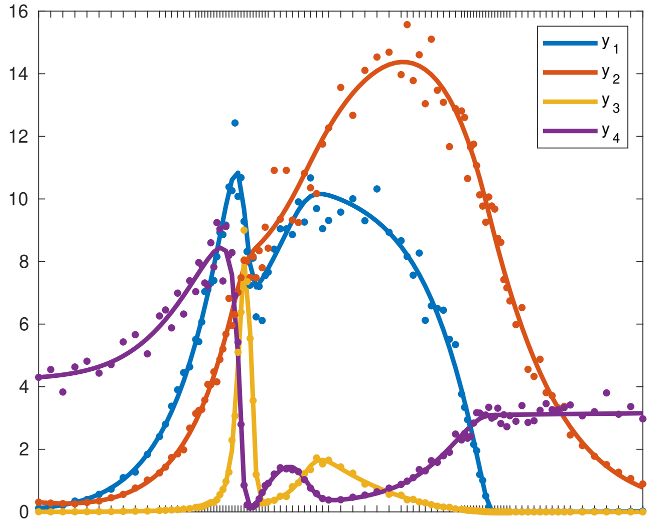

Figure 4.1: Queried points shown as ticks in the plot for Calcium Ion test problem

	PINN					
Test Problems	PINN-smoothed only		perturbed datapoints		progressive shooting	
	success	avg iters	success	avg iters	success	avg iters
Calcium Ion	42/100	93.5238	92/100	140.7935	85/100	315.7882
Mendes	100/100	24.9700	100/100	24.9700	100/100	46.7900
Barnes	100/100	4.1800	100/100	4.2500	100/100	3.9400
Goodwin	100/100	6.2400	100/100	6.0600	100/100	6.8100

Table 4.9: Results of running parameter estimation on 100 noisy datasets using PINN, with the nonlinear parameters fixed, on non-uniform grids.

	GP					
Test Problems	GP-smoothed only		perturbed datapoints		progressive shooting	
	success	avg iters	success	avg iters	success	avg iters
Calcium Ion	57/100	57.7719	100/100	92.3600	100/100	20.7700
Mendes	100/100	6.1300	100/100	6.1300	100/100	5.9900
Barnes	100/100	4.3100	100/100	3.3600	100/100	3.3300
Goodwin	100/100	6.1400	100/100	5.9800	100/100	5.0600

Table 4.10: Results of running parameter estimation on 100 noisy datasets using GP, with the nonlinear parameters fixed, on non-uniform grids.

	UKF					
Test Problems	UKF-refined p_0 only		perturbed p_0		progressive shooting	
	success	avg iters	success	avg iters	success	avg iters
Calcium Ion	89/100	29.4607	94/100	65.0000	100/100	13.7600
Mendes	99/100	8.1616	100/100	10.8300	99/100	8.1818
Barnes	100/100	4.2900	100/100	4.1700	100/100	4.2800
Goodwin	100/100	6.1700	100/100	6.1700	100/100	6.0000

Table 4.11: Results of running parameter estimation on 100 noisy datasets using UKF, with the nonlinear parameters fixed, on non-uniform grids. Note perturbed p_0's are generated by sampling from the estimated mean and covariance of the filtered p_0 in UKF.

Chapter 5

Discussion

With the two stage procedure, for each technique applied to each test problem, it seems that generally progressive shooting and datapoint perturbation are able to provide certain improvement on the experimental results in most cases, either increasing the success rate or lowering the computational effort required in the second stage. And progressive shooting gives the best results more often. The advantage of these two ways to provide a wider range of initial guesses could be less obvious in the non-uniform grid case, especially for datapoint perturbation in which the original initial guess by each method usually ends up being the best, rather than those obtained from the perturbed datapoints. That seems to be the reason for identical average number of iterations corresponding to the same technique in some test problem (looking at experimental results in the same row). Sometimes datapoint perturbation would give better results by having initial guess obtained on the perturbed trajectory, and other times those additional candidates are no better than the original initial guess. However, these candidates provide better chances of finding a good enough starting values especially when the initial guess obtained on the original observations is problematic, such as is the case of Calcium Ion problem.

PINN

One of the advantages of PINN is its potential applicability to a wider range of parameterized systems, due to the universal approximator property. For problems that have simpler trajectories, the method works well in terms of its ability to capture the true trajectory. However, it might not work well in problems such as the Calcium Ion problem, in which the shape of the trajectory is complicated. PINN does not also seem to improve as much with PS as some of the other methods, especially for the Calcium Ion problem. The reason is probably the poor approximation of the trajectory by PINN. In the Calcium Ion problem, the trajectory could exhibit multiple turning points or contains several peaks and jumps, etc. It seems to have some difficulty in capturing the true trajectory to a satisfactory level, especially near the end of the time interval. See Figure (5.1) for the plot of the smoothed curves against true trajectories and observed datapoints on uniform mesh. Even though PS does not seem to improve the results as much as the other methods, by considering observations from progressively increasing endpoints of the time interval in obtaining initial guesses, some of the initial guesses can exclude the worse approximations near the end of the time interval. If the objective function includes observations from the first half of the trajectory, which is usually better approximated, the result could be better, as evidenced in the increased success rates for PS.

Overall, although the setup of the PINN method seems to be general enough for arbitrary systems, it might not work well in every case and some tuning of the neural network structure is needed for each problem. Note also the result by running the PINN-estimated parameter as initial guess to second stage was not reported but similar behaviour could possibly be expected with the PINN technique.

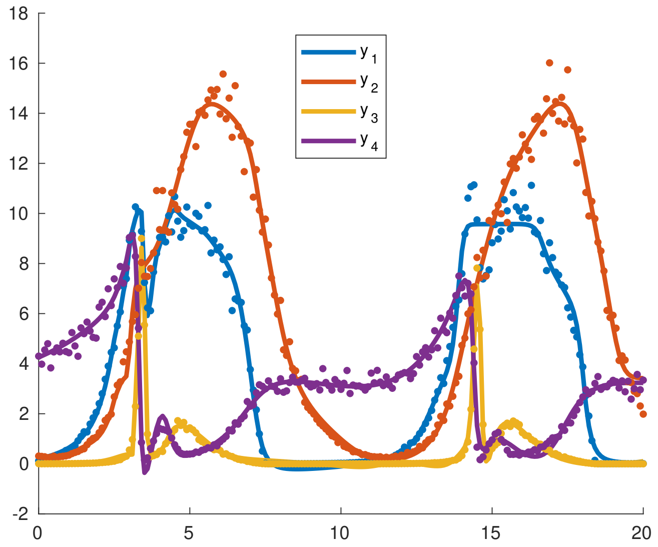

Figure 5.1: Plot of the smoothed curves obtained by PINN against the observed datapoints for the Calcium Ion test problem on a uniform mesh.

GP

Compared to PINN, the approximated curve by GP is better. See Figure (5.2) for the plot of the smoothed curves against the observed datapoints. It could act as an alternative way to provide a more reliable estimated underlying trajectory other than the original observations, and initial guesses obtained on both the smoothed as well as the perturbed curve can be kept as candidates.

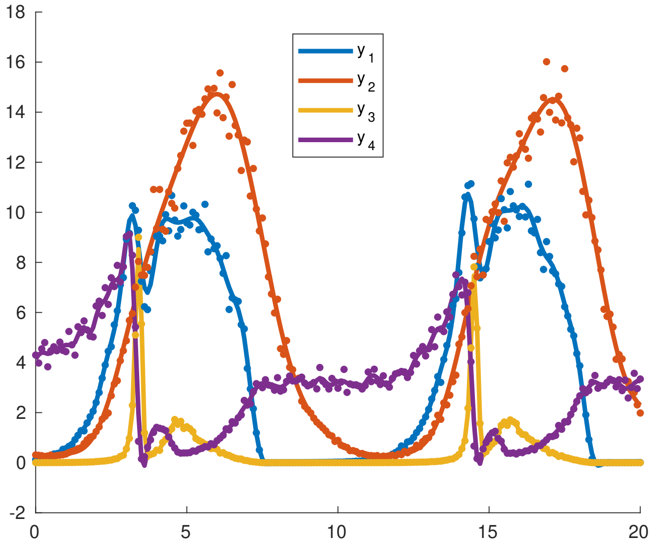

Figure 5.2: Plot of the smoothed curves obtained by GP against the observed datapoints for the Calcium Ion test problem on a uniform mesh.

UKF

UKF acts as a refinement and provides complementary initial guesses different from INT-SME. In terms of computational effort, the method makes the trade-off to spend more time in first stage for more accurate initial guess due to the simulation cost. It may suffer from the issue of occasional simulation failure, and occurrence of non-SPD matrices which will break the execution of the program. Therefore, before running UKF, it already requires a close enough initial guess, which sounds paradoxical to the goal of having UKF to find a good initial guess. Therefore in all our experiments, the INT-SME-generated initial guess is passed to the UKF procedure. In this linear parameter estimation case, at least, these possible failures were not observed. However, it would become more complicated if nonlinear parameters were to be estimated. Note it could also suffer from the difficulty of finding suboptimal local minimum. Additional perturbed parameters obtained from the estimated parameter probability distribution can be considered to help generate more initial guesses to start the second stage optimization. However, if the estimated mean of the parameters lies at a location far from the true means and the estimated variance is shrinking towards zero, the largely concentrated samples drawn from the badly-behaving mean will not significantly increase the chance of getting a good initial guess to start the second stage and one may still not be able to escape from a local minimum. Meanwhile, there is already a high cost associated with the simulation, preventing it from trying a large number of additional parameters. Also for the Mendes problem, we observed worse performance than using only INT-SME, which is expected since we try on only one pair of values for P and S, instead of all 16 pairs because of the trade-off to spend limited amount of time in the first stage. With 16 times fewer observations it seems likely that UKF will not result in an improved initial guess, compared to INT-SME itself.

For the implementation of datapoint perturbation, instead of always taking their estimated mean at the end of the time interval, we could also consider drawing samples from the estimated probabilistic distribution of parameters at other timesteps. This would later allow for multiple branches of filtered trajectories, depending on the different perturbed parameters used at different timesteps. But this will create additional exploration cost during the process of filtering.

Koopman

The Koopman approch is a flexible way of getting initial guess and it is relatively easy to implement. This technique could provide better initial guess if the curve is less corrupted by noise. Additional effort may need to be made in the investigation into the complex-valued matrix issue when the trajectory is corrupted by increasing magnitude of noise. Moreover, further investigation includes how to extend the method to 1) nonlinear parameter estimation, 2) non-uniform observation time points. PINN and GP seem to benefit more from perturbed datapoints while neither UKF nor Koopman seem to enjoy as much improvement from perturbed datapoints.

TLS

TLS is another approach to effectively obtain a more suitable initial guess. Its derivation differs from ordinary least square in that it aims to fix the inherent estimation bias caused by errors in the independent variables. The initial guess obtained with TLS improves the results by taking into consideration the noise assumption in the matrix constructed, instead of increasing the number of trials and relying on a collection of different initial guesses. TLS is able to significantly improve the results on the Calcium Ion problem and achieve comparable results to progressive shooting.

Future Work

Future work includes investigation into integrating weighted total least-squares (WTLS) with INT-SME, which considers imposing additional assumptions on the covariance between each independent variables. In addition, with the current formulation of the Koopman method, it only works with uniform grids. Future work may include extending the Koopman method to non-uniform grids as well as estimating nonlinear parameters. Future investigation might also include modifying GP, UKF to reflect the relative noise assumption.

Bibliography

[1] E. Balsa-Canto, M. Pfeifer, J. Banga, J. Timmer, and C. Fleck. Hybrid optimization method with general switching strategy for parameter estimation. *BMC Systems Biology*, 2(1), 2008.

[2] A. A. Berryman. The origins and evolution of predator-prey theory. *Ecology*, 73(5):1530–1535, 1992.

[3] M. Budišić, R. Mohr, and I. Mezić. Applied koopmanism. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 22(4):047510, 2012.

[4] B. Calderhead, M. Girolami, and N. D. Lawrence. Accelerating bayesian inference over nonlinear differential equations with gaussian processes. *Advances in Neural Information Processing Systems*, 21:217–224, 2009.

[5] J. Calver. Parameter estimation for systems of ordinary differential equations. *PhD thesis, Department of Computer Science, University of Toronto*, 2019.

[6] J. Calver, W.H. Enright, and J. Yao. Using shooting approaches to generate initial guesses for ode parameter estimation. In *International Conference on Applied Mathematics, Modeling and Computational Science 2019*, submitted.

[7] Y. Cao. Learning the unscented kalman filter. https://www.mathworks.com/matlabcentral/18217-learning-the-unscented-kalman-filter, 2010.

[8] I. Dattner and S. Gugushvili. Accelerated least squares estimation for systems of ordinary differential equations. *arXiv preprint arXiv:1503.07973*, 2015.

[9] S. T. M. Dawson, M. S. Hemati, M. O. Williams, and C. W. Rowley. Characterizing and correcting for the effect of sensor noise in the dynamic mode decomposition. *Experiments in Fluids*, 57(3):42, 2016.

[10] A. Gábor and J. R. Banga. Robust and efficient parameter estimation in dynamic models of biological systems. *BMC systems biology*, 9(1):74, 2015.

[11] A. Gelb. *Applied optimal estimation.* MIT press, 1974.

[12] G. H. Golub and C. F. van Loan. An analysis of the total least squares problem. *SIAM Journal on Numerical Analysis*, 17(6):883–893, 1980.

[13] D. Gonze and W. Abou-Jaoudé. The goodwin model: behind the hill function. *PloS ONE*, 8(8):445–466, 2013.

[14] B. C. Goodwin. Oscillatory behavior in enzymatic control processes. *Advances in enzyme regulation*, 3:425–438, 1965.

[15] S. Gugushvili, C.A.J. Klaassen, et al. $\sqrt{n}$-consistent parameter estimation for systems of ordinary differential equations: bypassing numerical integration via smoothing. *Bernoulli*, 18(3):1061–1098, 2012.

[16] K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

[17] S. J. Julier and J. K. Uhlmann. New extension of the Kalman filter to nonlinear systems. In *Signal Processing, Sensor Fusion, and Target Recognition VI*, volume 3068, pages 182 – 193. International Society for Optics and Photonics, 1997.

[18] R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(Series D):35–45, 1960.

[19] B. O. Koopman. Hamiltonian systems and transformation in hilbert space. *Proceedings of the National Academy of Sciences of the United States of America*, 17(5):315–318, 1931.

[20] F. T. Krogh, J. P. Keener, and W. H. Enright. Reducing the number of variational equations in the implementation of multiple shooting. *Numerical Boundary Value ODEs*, pages 121–135, 1985.

[21] U. Kummer, L. F. Olsen, C. J. Dixon, A. K. Green, E. Bornber-Bauer, and G. Baier. Switching from simple to complex oscillations in calcium signaling. *Biophysical Journal*, 79(3):1188–1195, 2000.

[22] A. Mauroy and J. Goncalves. Linear identification of nonlinear systems: A lifting technique based on the koopman operator. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, pages 6500–6505, Dec 2016.

[23] A. Mauroy and J. Goncalves. Koopman-based lifting techniques for nonlinear systems identification. *IEEE Transactions on Automatic Control*, 2019.

[24] C. Michalik, R. Hannemann, and W. Marquardt. Incremental single shooting - a robust method for the estimation of parameters in dynamical systems. *Computers & Chemical Engineering*, 33(7):1298 – 1305, 2009.

[25] C. G. Moles, P. Mendes, and J. R. Banga. Parameter estimation in biochemical pathways: a comparison of global optimization methods. *Genome research*, 13(11):2467–2474, 2003.

[26] M. Raissi, P. Perdikaris, and G. E. Karniadakis. Physics informed deep learning (part i): Data-driven solutions of nonlinear partial differential equations. *arXiv preprint arXiv:1711.10561*, 2017.

[27] M. Raissi, P. Perdikaris, and G. E. Karniadakis. Physics informed deep learning (part ii): Data-driven discovery of nonlinear partial differential equations. *arXiv preprint arXiv:1711.10566*, 2017.

[28] M. Rodriguez-Fernandez, J. A. Egea, and J. R. Banga. Novel metaheuristic for parameter estimation in nonlinear dynamic biological systems. *BMC Bioinformatics*, 7(1):1–18, 2006.

[29] M. Rodriguez-Fernandez, P. Mendes, and J. R. Banga. A hybrid approach for efficient and robust parameter estimation in biochemical pathways. *Biosystems*, 83(2):248–265, 2006.

[30] B. Schaffrin and A. Wieser. On weighted total least-squares adjustment for linear regression. *Journal of Geodesy*, 82(7):415–421, Jul 2008.

[31] A. Sitz, U. Schwarz, J. Kurths, and H. U. Voss. Estimation of parameters and unobserved components for nonlinear systems from noisy time series. *Physical review E*, 66(1):016210, 2002.

[32] 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.

[33] B. Wang and W. Enright. Parameter estimation for odes using a cross-entropy approach. *SIAM Journal on Scientific Computing*, 35(6):A2718–A2737, 2013.

[34] P. Wenk, G. Abbati, M. A. Osborne, B. Schölkopf, A. Krause, and S. Bauer. Odin: Ode-informed regression for parameter and state inference in time-continuous dynamical systems. *arXiv preprint arXiv:1902.06278*, 2019.

[35] P. Wenk, A. Gotovos, S. Bauer, N. Gorbach, A. Krause, and J. M. Buhmann. Fast gaussian process based gradient matching for parameter identification in systems of nonlinear odes. *arXiv preprint arXiv:1804.04378*, 2018.

[36] C. K. I. Williams and C. E. Rasmussen. *Gaussian processes for machine learning.* MIT press Cambridge, MA, 2006.

[37] M. O. Williams, I. G. Kevrekidis, and C. W. Rowley. A data–driven approximation of the koopman operator: Extending dynamic mode decomposition. *Journal of Nonlinear Science*, 25(6):1307–1346, 2015.

[38] M. O. Williams, C. W. Rowley, and I. G. Kevrekidis. A kernel-based approach to data-driven koopman spectral analysis. *arXiv preprint arXiv:1411.2260*, 2014.

[39] H. Zivari-Piran. Efficient simulation, accurate sensitivity analysis and reliable parameter estimation for delay differential equations. *PhD thesis, Department of Computer Science, University of Toronto*, 2009.