Modeling, Identification and Control, Vol. 40, No. 4, 2019, pp. 213–243, ISSN 1890–1328

Estimating uncertainty of model parameters obtained using numerical optimisation

O. M. Brastein [1] B. Lie [1] C. Pfeiffer [1] N.-O. Skeie [1]

[1] *Department of Electrical Engineering, Information Technology and Cybernetics, University of South-Eastern Norway, N-3918 Porsgrunn, Norway.*

Abstract

Obtaining accurate models that can predict the behaviour of dynamic systems is important for a variety of applications. Often, models contain parameters that are difficult to calculate from system descriptions. Hence, parameter estimation methods are important tools for creating dynamic system models. Almost all dynamic system models contain uncertainty, either epistemic, due to simplifications in the model, or aleatoric, due to inherent randomness in physical effects such as measurement noise. Hence, obtaining an estimate for the uncertainty of the estimated parameters, typically in the form of confidence limits, is an important part of any statistically solid estimation procedure. Some uncertainty estimation methods can also be used to analyse the practical and structural identifiability of the parameters, as well as parameter inter-dependency and the presence of local minima in the objective function. In this paper, selected methods for estimation and analysis of parameters are reviewed. The methods are compared and demonstrated on the basis of both simulated and real world calibration data for two different case models. Recommendations are given for what applications each of the methods are suitable for. Further, differences in requirements for system excitation are discussed for each of the methods. Finally, a novel adaption of the Profile Likelihood method applied to a moving window is used to test the consistency of dynamic information in the calibration data for a particular model structure.

Keywords: Parameter estimation, Uncertainty analysis, Bootstrapping, Profile Likelihood

1 Introduction

1.1 Background

1.1.1 Dynamic system models

Dynamic system models are important for a large range of scientific and industrial applications, such as *model predictive control* of dynamic systems Killian and Kozek [2016], Wang [2009] or creating *digital twins* of chemical process plants for monitoring or operator training Rosen et al. [2015]. Typically, the performance of the overall system depends on the accuracy of the model predictions. Often, models contain parameters that are difficult to obtain from system specifications. Hence, calibration of model parameters is an important part of developing good quality dynamic models. Additionally, the model parameters are sometimes used as *soft-sensors* for system variables that are otherwise difficult to measure. This requires a specific physical interpretation of the estimated parameters, which places further requirements on the model calibration process.

For many real world processes, models can be created based on balance laws and application of detailed knowledge about the physics and chemistry involved in the process. This approach often includes approximations in order to keep the model's detail level manageable. Such models are classified as *mechanistic,* or *white-box*, since they describe detailed physical information about the system mechanisms, using a mathematical language, in a way that is interpretable by a

doi:10.4173/mic.2019.4.3

human expert. For this type of models, parameters tend to be derived from physical specifications of the system. It is also common for such models to include parameters that require estimation from measurement data.

An alternative approach to creating dynamic system models is the use of *System Identification* (SID) methods Ergon and Di Ruscio [1997], Ljung [1999], where models are created by calibrating parameters of a *predetermined* mathematical structure in order to capture the relevant dynamic system behaviour. SID models are created without explicit use of prior physical information, hence, they are often classified as *black-box,* or *data-driven,* models Kristensen et al. [2004]. One advantage of the SID approach is that it captures the process behaviour directly from measurement data, which avoids modelling errors caused by *insufficient specification* of the system. To properly capture the system behaviour, SID methods require a complete set of basis functions Farrell and Polycarpou [2006]. If the applied set of basis functions is insufficient, the identified model may still *approximate* the system behaviour, but with model errors, e.g., non-linear system identified using a linear model structure. Further, SID methods obtain all system information from data, hence the quality of the *calibration* data, in particular the level of *dynamic information*, directly influences the quality of the obtained model. Finally, the SID approach tends to provide better statistics on the model accuracy, produced as part of the calibration procedure Johansson [1993], Ljung [1999].

A third, intermediate, possibility is to combine *cognitively* constructed model structures, based on *naive* prior physical knowledge, with parameter calibration, to create a *simplified* lumped parameter model. The resulting model, often classified as *grey-box*, tends to have most, if not all, its parameters unknown, which requires full model calibration Berthou et al. [2014], Bohlin and Graebe [1995], Kristensen et al. [2004]. Due to the significant approximations applied in the creation of grey-box models, they should be treated in a stochastic framework, using *Stochastic Differential Equations (SDE)* to describe the dynamic system behaviour. These models are *approximations* by design, using only limited physical insight, which introduces significant *epistemic* uncertainty. However, since the models are based on, at least, a *naive* physical understanding of the underlying system, the parameters are often *assumed* to be physical constants.

Arguably, most white-box models contain some uncertainty in the formulation, which gives rise to model errors, and can therefore benefit from application of grey-box modelling methods for parameter estimation. This approach has indeed been claimed as a natural framework for modelling dynamic systems in general Bohlin and Graebe [1995], Kristensen et al. [2004].

1.1.2 Identifiability

Parameters of models derived, at least partially, from prior knowledge of the underlying physical system are often *assumed* to be constants of the system. Subsequently, a globally optimal value, which can be obtained *unambiguously* by optimisation, is assumed to exist. This assumption should, however, be verified in the context of *parameter identifiability* Ferrero et al. [2006], Johansson [1993], Juhl et al. [2016a], Raue et al. [2009]. This is especially important for *grey-box* models, which contain large *epistemic* uncertainty due to the strong approximation applied in their construction.

It is well known that models can contain parameters that are *structurally* non-identifiable due to over-parametrisation, which leads to parameter redundancy, or parameters for which perturbations of the parameter values have no observable effect on the model output Ferrero et al. [2006], Johansson [1993], Raue et al. [2009]. Additionally, lack of *sufficient excitation* of the system during data acquisition may lead to *practical* non-identifiability of certain parameters Deconinck and Roels [2017], Ferrero et al. [2006], Johansson [1993], Murphy and Van der Vaart [2000], Raue et al. [2009]. If the measured inputs and outputs of the physical system do not contain the necessary dynamic information, the influence of some parameters on the error function used for parameter optimisation may be negligible, thus leading to non-identifiability. While *structural* identifiability is independent of the experimental conditions, *practical* identifiability is a function of the dynamic information content in the data-set, and subsequently depends on the experimental configuration Raue et al. [2009].

Due to these potential challenges with parameter identifiability, a model structure may be *designed* with parameters that are *intended* to have a specific physical meaning, but it is not certain that the *estimated* parameters support this assumption Deconinck and Roels [2017]. While the parameters of *physical* systems are clearly *constants* of the system, the *estimated* parameters of a model are always subject to uncertainty and potential non-identifiability.

1.2 Previous work

1.2.1 Parameter estimation and the CTSM framework

Estimation of parameters requires a well defined objective function which *adequately* describes the model fit. Several alternatives are used in the literature, such

as the shooting/ballistic simulation error approach, based on deterministic simulations Berthou et al. [2014], Brastein et al. [2018], or the *maximum likelihood* approach used in the Continuous Time Stochastic Modelling (CTSM) framework Kristensen et al. [2004], Madsen and Holst [1995]. CTSM is based on maximising the *likelihood function Akaike [1998], Rossi [2018]* evaluated by computing residuals, which are assumed to be Normal distributed, in a Kalman Filter. This method has previously been developed in a number of publications Bacher and Madsen [2011], Juhl et al. [2016b], Kristensen and Madsen [2003], Kristensen et al. [2004], Madsen and Holst [1995] and implemented in the CTSM framework Kristensen and Madsen [2003]. This approach offers the advantage of an objective function with a solid statistical framework, which enables use of statistical tools for model validation and selection Kristensen et al. [2004].

1.2.2 Profile likelihood

While *structural* identifiability is well defined in the literature, the *practical* identifiability of parameters is less clearly defined Raue et al. [2009]. Although there are several methods that can identify structural non-identifiability, e.g., Power Series Expansion Pohjanpalo [1978], it is desirable to have a method that can identify *both* types of parameter identifiability. A good choice is the *Profile Likelihood* (PL) method, which creates profiles or distributions of the parameter likelihood, and subsequently can produce likelihood based confidence intervals Deconinck and Roels [2017], Murphy and Van der Vaart [2000], Raue et al. [2009], Venzon and Moolgavkar [1988]. These intervals can be used to diagnose parameter identifiability Raue et al. [2009].

1.2.3 Bootstrapping for time-series data

The idea of Bootstrapping was first introduced in Efron [1979], as a method of estimating the *variance*, i.e., uncertainty, of a statistic. The method has become popular, due in part to its simplicity. The fundamental idea in bootstrapping is to estimate *properties*, such as the *uncertainty* of an estimated parameter, by *randomly drawing samples with replacement* from the *original* data, thus obtaining multiple *different* data-sets. These different data-sets will produce *slightly* different parameter estimates, which allows estimation of the *uncertainty* of the estimated parameter by computing the *mean* and *covariance* of the bootstrapped estimates. Data-sets generated by bootstrapping are often called *pseudo* data-sets to emphasise the fact that they are all re-combinations of the original data, and *not, new, independent* data-sets collected from the physical system. An interesting property of the bootstrapping method is its intuitive similarity to the basis of the confidence interval (CI) as presented in Kullback [1939], Neyman [1937]; running *multiple* experiments to compute the uncertainty of results.

A fundamental requirement of the bootstrap method, as presented in Efron [1979], is that the samples in the original data must be *independently and identically distributed* (i.i.d), which is a property not usually observed for time-series data Kunsch [1989]. Hence, there has been several adaptions of bootstrapping for time-series data. One solution is to fit a parametric Auto Regressive Moving Average (ARMA) model to the data, and bootstrap the residuals, which are presumed i.i.d, to create new data-sets Kunsch [1989], Lie [2009], Politis [2003]. However, this approach is limited to systems which can be adequately described by such model structures and thus produce i.i.d. residuals. Hence, *non-parametric* approaches to bootstrapping for time-series data has been receiving significant interest in research Kunsch [1989], Lodhi and Gilbert [2011], Politis [2003], Politis and Romano [1994]. In particular, various forms of block based bootstrapping, i.e., methods that segment the data into blocks, and draw randomly with replacements from the blocks, rather than the samples themself, has shown promising results Kunsch [1989]. Examples include overlapping and non-overlapping block bootstrap Kunsch [1989], Lodhi and Gilbert [2011], moving block bootstrap Kunsch [1989], and stationary bootstrapping Politis and Romano [1994]. For a detailed review of bootstrapping for time-series data, see Politis [2003].

1.3 Overview of paper

In this paper, selected methods for estimating uncertainty of estimated parameters are presented and demonstrated on two separate test cases. The methods discussed in this paper are based on the use of *numerical optimisation*, using a well defined objective function to evaluate the fit of a parameter set. The focus in this paper is on analysing the *parameters* themselves, rather than the prediction accuracy of the calibrated model.

The theoretical foundation of parameter estimation and analysis is presented in Section 2. Sections 2.1 and 2.2 detail the foundation of parameter estimation and the representation of uncertainty, respectively. Section 2.3 presents the theoretical foundation of each of the discussed methods. Results of applying the methods is presented in Section 3. Finally, recommendations as to what applications each of the methods is most suitable for are given in Section 3.3 before the paper is concluded in Section 4.

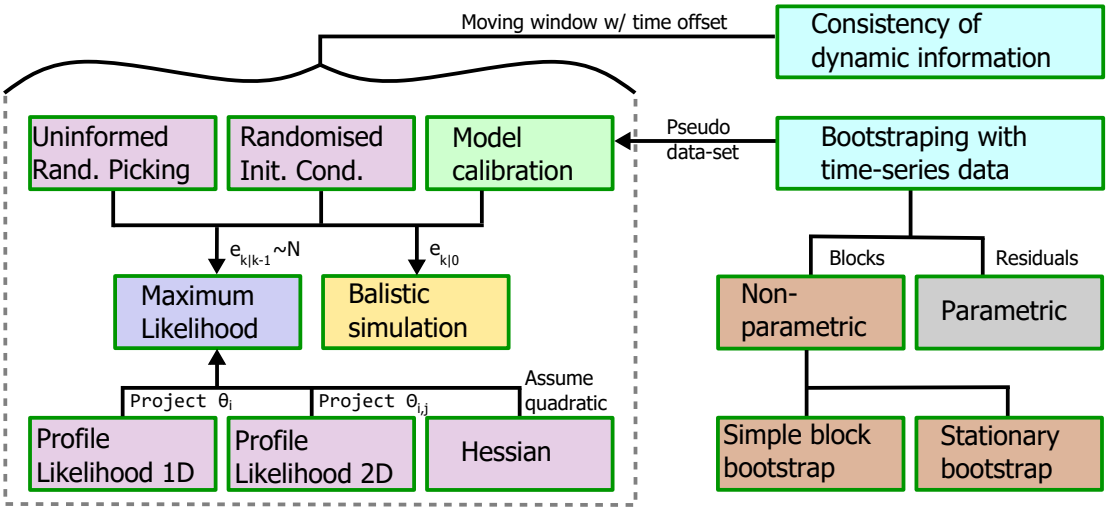

Figure 1: An overview of all the methods discussed in this paper, and how they are related. On the left-hand side, six optimisation based methods are shown, and related with the two types of objective functions that are discussed. The top three methods: Uninformed Random Picking, Randomised Initial Conditions and straight forward Model Calibration, can be used with both types of objective function. The lower three methods: Profile Likelihood 1D and 2D, and the Hessian curvature method, all require a Maximum Likelihood objective function. The top right block in the figure shows the moving window method, which can be used with all the six different methods to test the results for consistency in time. Next, the discussed variations of bootstrapping for time-series data are shown, and associated with model calibration, since the working principle of bootstrapping methods requires separate calibration of parameters for each generated pseudo data-set. Further, the two types of bootstrapping for time-series data, parametric vs non-parametric, are shown. Only non-parametric bootstrapping is discussed in this paper, specifically simple block bootstrapping and stationary bootstrapping, as illustrated in the lower right part of the figure.

2 Theoretical basis

This section discusses in detail several methods for estimating parameters of dynamic models, and, in particular, methods that also estimate the *uncertainty* and *identifiability* of the estimated parameters. An overview of the methods is presented in Fig. 1.

2.1 Parameter estimation

Estimation of parameters θ for a known model structure $\mathcal{M}$ can formally be defined as solving the optimisation problem:

$$\hat{\theta} = \arg\min_{\theta} g(\theta; \mathcal{M}, \mathcal{K}, \mathcal{A}) \quad (1)$$
$$\text{s.t. } \theta \in \Theta$$

Here, $\Theta = \{\theta : \theta_{\min} < \theta < \theta_{\max}; \theta \in \mathbb{R}^{n_\theta}\}$ is a continuous space of feasible values for the model parameters, where $\theta_{\min}$ and $\theta_{\max}$ are the lower and upper bounds. Hence, the space Θ forms inequality constraints for the optimisation problem in Eq. (1). $\mathcal{K}$ are the experimental conditions, including a set of measurements of system inputs and outputs, which are defined in continuous time as input $u_t \in \mathbb{R}^{n_u}$ and output $y_t \in \mathbb{R}^{n_y}$, and the corresponding ordered sequences of discrete time measurements u_k and y_k:

$$y_{[N]} = [y_0, y_1, \ldots, y_N] \quad (2)$$
$$u_{[N]} = [u_0, u_1, \ldots, u_N]$$

Here, the integer subscripts $k = 0, 1, \ldots, N$ denote the discrete time sampling instants and the *subscript* enclosed in $[\cdot]$ is used to indicate an ordered sequence. These measurements are used to evaluate the objective function $g(\theta)$ when θ is varied over the feasible set Θ by a numerical optimisation algorithm $\mathcal{A}$.

2.1.1 Uncertainty of estimated parameters

A common assumption is that $\mathcal{S} \in \mathcal{M}(\Theta)$, where $\mathcal{S}$ is the true system, and that there exists a true parameter vector θ^* such that $\mathcal{M}(\theta^*) \equiv \mathcal{S}$. However, if the data used for parameter estimation, i.e., $y_{[N]}$ and $u_{[N]}$, is collected as *measurements* from the physical system $\mathcal{S}$, it will contain *aleatoric* uncertainty

due to the inherent randomness of *measurement noise* Bentley [2005], Ljung [1999]. Additionally, most dynamic system models contain approximations with respect to $\mathcal{S}$, which also introduces some *epistemic* uncertainty in the model equations. Hence, the assumption $\mathcal{M}(\theta^*) \equiv \mathcal{S}$ is almost always questionable. Further, the estimate $\hat{\theta}$ depends on the amount of dynamic information in $\mathcal{K}$, the choice of model fit objective function $g(\theta)$, and to some extent on the optimisation algorithm $\mathcal{A}$. Hence, even if there exists a well defined, globally optimal, set of parameters θ^*, it may not be possible to obtain an *unambiguous* parameter *estimate*. Therefore, prior to interpreting a set of estimated parameters as *determined* by the physical properties of $\mathcal{S}$, and subsequently assuming a physical interpretation to the *estimated* parameter values, it is necessary to analyse the estimation uncertainty and *identifiability* of the parameters Ljung [1999].

2.1.2 Stochastic parameter estimation

Dynamic system models are typically formulated with uncertainty in both the *state transition* and *measurement* equations Bohlin and Graebe [1995], Kristensen et al. [2004], Simon [2006]. Such models can conveniently be expressed as a continuous time stochastic differential equation (SDE) for the state transition Jazwinski [1970]. Since the data used for calibration is typically available only at discrete time instants, a discrete time measurement equation is a convenient formulation. Adopting the notation of Kristensen et al. [2004]:

$$dx_t = f(x_t, u_t, t, \theta)\,dt + \sigma(u_t, t, \theta)\,d\omega_t \tag{3}$$

$$y_k = h(x_k, u_k, t_k, \theta) + e_k \tag{4}$$

where $t \in \mathbb{R}$ is the time variable and $x_t \in \mathbb{R}^{n_x}$ is the continuous time state vector. The first and second terms in Eq. (3) are commonly called the *drift* and *diffusion* term, respectively Jazwinski [1970], Kristensen et al. [2004]. The diffusion term expresses the process noise as the function σ multiplied with the differential of a standard Wiener process ω_t Jazwinski [1970], Kristensen et al. [2004]. The discrete time *measurement equation* is given in Eq. (4). The CTSM framework Kristensen and Madsen [2003], Kristensen et al. [2004], Madsen and Holst [1995] presents a statistically solid approach to estimating parameters in such stochastic models. A *Maximum Likelihood* estimate of θ can be obtained by deriving the objective $g(\theta)$ in Eq. (1) from the likelihood $L(\theta)$, defined as the *joint* probability $\mathrm{P_r}(\cdot)$ of observing the measurement sequence $y_{[N]}$ when θ and $\mathcal{M}$ are known, i.e., $L(\theta; y_{[N]}, \mathcal{M}) = \mathrm{P_r}(y_{[N]}|\theta, \mathcal{M})$. Note that while $L(\theta; y_{[N]}, \mathcal{M})$ is defined using probability, the resulting likelihood function is *not* a probability *distribution*, since the integral of the likelihood over all possible parameters does *not* equal 1.

For simplicity of notation, the model structure $\mathcal{M}$ is implicitly assumed known and omitted from the condition. The likelihood can be expanded to conditional probabilities by the chain rule:

$$L(\theta; y_{[N]}) = \left(\prod_{k=1}^{N} \mathrm{P_r}(y_k|y_{[k-1]}, \theta)\right) \mathrm{P_r}(y_0|\theta) \tag{5}$$

Equation (3) assumes the diffusion term to be additive and independent of the state x, and driven by a *Wiener process* Kristensen et al. [2004]. Hence, it is reasonable to assume that the conditional probabilities in Eq. (5) can be approximated by Gaussian distributions Kristensen and Madsen [2003], Kristensen et al. [2004]. The likelihood can then be expressed as a multivariate Gaussian distribution Kristensen et al. [2004],

$$L(\theta; y_{[N]}) = \left(\prod_{k=1}^{N} \frac{\exp\left(-\frac{1}{2}\epsilon_{k|k-1}^T \mathcal{E}_{k|k-1}^{-1} \epsilon_{k|k-1}\right)}{\sqrt{\det(\mathcal{E}_{k|k-1})}\,(\sqrt{2\pi})^{n_y}}\right) \mathrm{P_r}(y_0|\theta) \tag{6}$$

To ensure that Eq. (6) is justified, the normality assumption on the residuals can, and *should*, be checked during model validation Johansson [1993], Kristensen et al. [2004]. Model validation is further discussed in Section 2.1.4

The residuals $\epsilon_{k|k-1}$ and their covariance $\mathcal{E}_{k|k-1}$ are needed to evaluate Eq. (6). These quantities can be obtained by use of a Kalman Filter (KF):

$$\hat{y}_{k|k-1} = \mathbb{E}\left[y_k|y_{[k-1]}, \theta\right] \tag{7}$$

$$\epsilon_{k|k-1} = y_k - \hat{y}_{k|k-1} \tag{8}$$

$$\mathcal{E}_{k|k-1} = \mathbb{E}\left[\epsilon_{k|k-1}\epsilon_{k|k-1}^T\right] \tag{9}$$

The choice of KF implementation, either the standard linear KF for linear models, or a non-linear variant such as the *Extended KF* (EKF) or the *Unscented KF* (UKF), depends on the model equations Brastein et al. [2019a].

Equation (6) is further simplified by conditioning on knowing y_0, taking the negative logarithm, and eliminating the factor $\frac{1}{2}$. Finally, the objective $g(\theta)$ in Eq. (1) is defined as $g(\theta; \mathcal{M}, \mathcal{K}) = \ell(\theta)$ where the log likelihood function $\ell(\theta)$, omitting the dependency on $y_{[N]}$ for simplicity of notation, is given as

$$\ell(\theta) = \sum_{k=1}^{N} \epsilon_{k|k-1}^T \mathcal{E}_{k|k-1}^{-1} \epsilon_{k|k-1} + \ln\left(\det\left(\mathcal{E}_{k|k-1}\right)\right) \tag{10}$$

The constant term $c = N \cdot n_y \cdot \ln(2\pi)$ has also been omitted.

2.1.3 Deterministic parameter estimation

Dynamic system models typically contain both *aleatoric* and *epistemic* uncertainty caused by the inherent randomness of measurements and the use of approximations in the model equations, respectively. Despite the well understood stochastic nature of such models, it is a common practice to treat all uncertainty as *aleatoric* and present at the model output. This results in a *deterministic*, sometimes called a *shooting* or *ballistic,* simulation, approach, in which the simulated state trajectory is *completely determined* by the given parameter vector θ, the initial conditions, and the measured system inputs. Essentially, the parameter estimation problem is then formulated as a *curve fitting* of the state trajectory transformed through the measurement equation. Rewriting the model from Eqs. (3) and (4) in discrete time *without* the diffusion term, let

$$\begin{aligned} \hat{x}_{k|0} &= \tilde{f}\left(\hat{x}_{k-1|0}, u_k, \theta\right) \\ \hat{y}_{k|0} &= \tilde{h}\left(\hat{x}_{k|0}, u_k, \theta\right) + e_k \end{aligned} \tag{11}$$

where the estimated state $\hat{x}_{k|0}$, and subsequently the estimated output $\hat{y}_{k|0}$, at time k are computed using *only* information available at *initial* time. The *Ordinary Least Squares* (OLS) estimate of the parameters is obtained by minimising the sum of square errors (SSE):

$$\tilde{g}\left(\theta\right) = \sum_{k=1}^{N} \tilde{\epsilon}_{k|0}^{T} Q \tilde{\epsilon}_{k|0} \tag{12}$$

where Q is a weighting matrix. Here, the estimation error $\tilde{\epsilon}_{k|0} = y_k - \hat{y}_{k|0}$ depends only on information at initial time t_0, which is in contrast to the residual obtained by *the one-step ahead predictions* in Eq. (8).

It is interesting to observe that the estimate obtained by minimising Eq. (12) corresponds to the *maximum likelihood estimate* (MLE) obtained from minimising Eq. (6) if, and only if, $\tilde{\epsilon}_{k|0} = \epsilon_{k|k-1}$ and the innovation covariance $\mathcal{E}_{k|k-1}$ is constant such that $Q = \mathcal{E}^{-1}$. Hence, minimising Eq. (12) gives an MLE estimate of the parameters only if the state transition model is *exact* w.r.t. the data generating system $\mathcal{S}$, i.e., the uncertainty associated with the diffusion term in Eq. (3) is zero and the measurement noise distribution is *stationary* with zero mean. Note also that if all measurements have the same variance, i.e., $\mathcal{E}^{-1} = Q = c \cdot I$ in Eq. (12), the weighting matrix can be taken outside the summation and subsequently eliminated, thus obtaining the *unweighted* least squares estimate.

While this can be a reasonable approximation, it is rarely exactly true, except when the calibration data is generated by simulations of the same model structure $\mathcal{M}$. Observe further that, assuming affine noise, this corresponds to obtaining the quantities in Eqs. (7) to (9) in a Kalman Filter with the process noise covariance $\mathcal{W} = 0$ and constant measurement noise covariance $\mathcal{V}$. Hence, the deterministic shooting error approach to parameter estimation may be seen as a special case of the scheme used in the CTSM framework and outlined in Section 2.1.2.

An interesting observation from comparing the two types of error calculation, e.g., $\tilde{\epsilon}_{k|0} = y_k - \hat{y}_{k|0}$ and $\epsilon_{k|k-1} = y_k - \hat{y}_{k|k-1}$, is that the SSE objective computed based on $\tilde{\epsilon}_{k|0}$ in Eq. (12) will have a gradient that is strongly non-linear in the parameters, due to the recursive predictor used in Eq. (11), i.e., $\hat{y}_{k|0} = f\left(\hat{y}_{k-1|0}, u_{k-1}, \theta\right)$. In contrast, the one-step-ahead prediction based likelihood objective in Eq. (10) will have a gradient that is linear in the parameters, since the predictor for the output is a function of measurements at previous time-steps, i.e., $\hat{y}_{k|k-1} = f\left(y_{[k-1]}, u_{k-1}, \theta\right)$.

2.1.4 Model validation

Since the objective function $\ell\left(\theta\right)$ in Eq. (10) depends on an assumption of normally distributed residuals, computed from one-step ahead predictions in a KF, it is necessary to verify the normality assumption subsequent to estimating model parameters. The literature detailing the CTSM framework specifically calls for evaluation of the residuals to verify the normality assumption Kristensen and Madsen [2003], Kristensen et al. [2004]. A practical test for normality can be applied by computing and plotting a cumulative periodogram (CP) of the residuals Deconinck and Roels [2017], Kristensen and Madsen [2003], Kristensen et al. [2004], where the *Kolmogorov-Smirnov criterion* can be used to place confidence bounds on the CP test Madsen [2007]. There are also a number of alternative tests for normality that can be applied such as the *zero-crossings* test or the *Kolmogorov-Smirnov* test Johansson [1993].

The possibility of validating a dynamic system model by testing the residuals for normality is a distinct *advantage* of the stochastic parameter estimation framework. For a *deterministic shooting simulation* approach, in which there may be *bias* errors that carry over from the state estimate at the previous time-step as shown in Eq. (11), there can be no reasonable assumption of normality for the estimation error $\tilde{\epsilon}_{k|0}$, unless the state transition model is *exact Madsen [2007].*

2.2 Expressing uncertainty of estimated parameters

A convenient way of describing the uncertainty of estimated parameters is by defining a sub-region in Θ, with some specific statistical criteria quantifying the

uncertainty of the parameters in the sub-region relative to the true, but unknown, parameters θ^*. One possible choice is the use of a *confidence region* with stated confidence α Neyman [1937], Raue et al. [2009]. In general, a region of arbitrary shape in Θ can be defined as a set, based on the *difference* in the objective function *relative* to a presumed optimal estimate $\hat{\theta}$:

$$\left\{\theta : g(\theta) - g\left(\hat{\theta}\right) < \Delta\right\} \tag{13}$$

where the threshold Δ is defined by some appropriate *statistical criterion*. The definition of the threshold depends on how the objective function g is defined, e.g., for a likelihood objective the thresholds can be computed from the χ^2 distribution as shown in Section 2.2.2. The set in Eq.(13), which contains any parameters θ for which the objective differs from the optimum by less than Δ, can be of any shape, including multimodal. However, the computation of a free-form set will require a large number of evaluations of the objective function for different θ in order to determine the set members. Therefore, a common approximation is to assume an ellipsoid, rather than free-form, region, defined as

$$\left\{\theta : \left(\theta - \hat{\theta}\right)^T \Sigma^{-1} \left(\theta - \hat{\theta}\right) < \Delta\right\} \tag{14}$$

where the *size* of the ellipsoid is determined by the threshold Δ, again computed by some appropriate statistical criterion. The weighting matrix Σ, typically the covariance of the estimated parameters, determines the *rotation* and *relative length* of the ellipsoid axes. Regions defined as in Eq. (14) also define a set of θ based on relative deviation compared to $\hat{\theta}$, but by assuming a quadratic approximation, the ellipsoid region is much faster to compute.

The points on the ellipsoid surface can be obtained by utilising the Cholesky decomposition $\Sigma = LL^T$, assuming Σ is positive definite Press et al. [2007]:

$$\left(\theta - \hat{\theta}\right)^T \Sigma^{-1} \left(\theta - \hat{\theta}\right) = \Delta \Rightarrow \left|L^{-1}\left(\theta - \hat{\theta}\right)\right|^2 = \Delta \tag{15}$$

Next, suppose x is a point on a unit hypersphere, then the ellipsoid surface boundary is obtained by the affine transformation

$$\theta = \hat{\theta} + \sqrt{\Delta} L x \tag{16}$$

2.2.1 Asymptotic confidence regions

Two common ways of expressing uncertainty is by defining a region in Θ, either a *univariate, point-wise,* confidence interval (CI), or a *multivariate, simultaneous,* confidence region, both defined by their prescribed confidence level α Neyman [1937].

Asymptotic CIs are based on the curvature of the objective function, which can be computed by utilising the covariance Σ_θ of the estimated parameters around the optimum $\hat{\theta}$ Deconinck and Roels [2017], Raue et al. [2009] to define a region on the form in Eq. (14). The threshold is then $\Delta = \Delta_\alpha$, where $\Delta_\alpha$is computed from the $\chi^2_{\alpha,n_{\text{df}}}$ distribution, with degrees of freedom n_{df} equal to the number of parameters in the *simultaneous* confidence region Press et al. [2007]. Observe that for *point-wise* confidence intervals of single parameters, Eq. (16) with $x \in \{\cos(0), \cos(\pi)\} = \{1, -1\}$ reduces to the familiar confidence interval for a scalar variable, where $\Sigma_{i,i} = \sigma_i^2$ Raue et al. [2009], i.e.;

$$\hat{\theta}_i \pm \sqrt{\Delta_\alpha \Sigma_{i,i}} \tag{17}$$

For *point-wise* intervals, Δ_α is drawn from the $\chi^2_{\alpha,n_{\text{df}}}$ distribution with $n_{\text{df}} = 1$ which is equivalent to the Normal c.d.f. with $\alpha/2$ confidence in each tail. The use of asymptotic confidence regions is widespread in all branches of science, particularly due to their ease of computation. If the *parameters* are in fact Gaussian distributed, the ellipsoid confidence regions are exact which further strengthens their popularity.

2.2.2 Likelihood based confidence regions

Unlike the asymptotic confidence interval in Eq. (17), a *likelihood based confidence interval* is computed by applying a *threshold* on the likelihood function to compute a confidence region in the form Eq. (13) Meeker and Escobar [1995], Raue et al. [2009]. Let

$$\left\{\theta : \ell(\theta) - \ell\left(\hat{\theta}\right) < \Delta_\alpha\right\} \quad , \quad \Delta_\alpha = \chi^2_{\alpha,n_{\text{df}}} \tag{18}$$

where $\hat{\theta}$ is a freely estimated parameter vector, which is presumed optimal, and the threshold Δ_α is the α percentile of the $\chi^2_{\alpha,n_{\text{df}}}$-distribution with n_{df} degrees of freedom. It follows from *Wilks' theorem* Wilks [1938] on the logarithm of the likelihood ratio Λ that the test statistic

$$2\ln(\Lambda) = 2\ln\left(\frac{L(\theta)}{L\left(\hat{\theta}\right)}\right) = \ell(\theta) - \ell\left(\hat{\theta}\right)$$

can be used to compare two models. The difference in log likelihood $\ell(\theta) - \ell\left(\hat{\theta}\right)$ is asymptotically χ^2-distributed Meeker and Escobar [1995], Raue et al. [2009], with n_{df} equal to the difference in number of free parameters between θ and $\hat{\theta}$ Press et al. [2007].

Arguably, likelihood based confidence intervals are conceptually simpler than asymptotic CIs due to their thresholded set definition. However, determining the set members is computationally intensive. An advantage of the likelihood based CI is that, due to its set

form definition, it does not assume a symmetric distribution of the parameters, and can in fact take on *any* shape, including multi-modal. Hence, likelihood based CIs are often considered superior to asymptotic CIs Raue et al. [2009].

2.2.3 Parameter profiles or distributions

An alternative to presenting the uncertainty of the estimated parameters as regions in Θ is to present the parameters as a *distribution* in Θ. Typically, a statistical quantity is used to create the profile, such as a *probability density function* or the *log-likelihood*. Profiles can be created over the entire Θ, or a subset of Θ as projections to single parameters θ_i, or planes $\Theta_{i,j}=\{\theta_i, \theta_j\}$, such that $\Theta_{i,j} \subset \Theta$. A parameter profile is more *descriptive* than a confidence region, since it shows how the uncertainty is *distributed* across the parameter space Θ. Since parameter profiles can be converted to confidence regions by applying some statistically defined threshold, they may be considered a superior form of uncertainty description. An example of this approach is the *Profile Likelihood* method presented in Section 2.3.4.

Another method of obtaining distributions of parameters, which are in fact probability distributions for the parameter θ, is through the use of Bayesian statistics and *Markov Chain Monte Carlo* (MCMC) methods. However, these methods are beyond the scope of this paper.

2.2.4 Interpretation of confidence regions

An interesting observation relating to the *interpretation* of the computed regions is that, while quite often assumed in published literature, the *confidence* of the computed regions is *not* a statement on the *probability* of said region containing the true parameters θ^*, as clearly stated in Neyman [1937]. Both θ^* and the computed confidence region are constants, not random variables. Hence, their relationship is not a question of probability, except for the trivial values of *zero* and *one,* which simply state whether or not the true parameter is a *member* of the computed confidence region. However, what *can* be stated in probabilistic terms is the *expected* probability of capturing θ^* in the CI, *prior* to performing the experiment and computing the interval, which is equal to the confidence α Kullback [1939]. This *expected* probability of capturing θ^* is called the *coverage probability*. If multiple experiments are carried out, with subsequent computations of CIs, the *ratio* of intervals that successfully captures the true parameter θ^* to the total number of experiments performed will be equal to the *coverage probability* Kullback [1939].

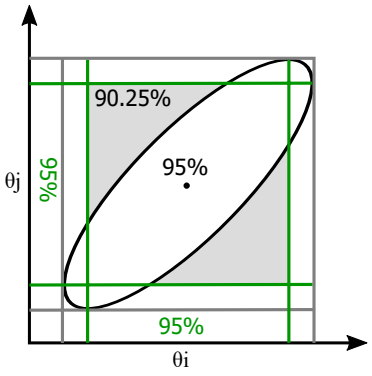

Figure 2: Comparing a simultaneous confidence ellipse for two parameters with the point-wise CIs (green) for each parameter, shows that the projections of the ellipse (grey) is wider than the point-wise CIs. Also, the combined confidence ($0.95\times0.95=0.9025$) of both point-wise CIs together is shaded in grey. Note the difference between two combined point-wise CIs (shaded square) and the simultaneous confidence ellipse.

2.2.5 Simultaneous and point-wise confidence regions

As discussed in Sections 2.2.1 and 2.2.2, the uncertainty of estimated parameters can be expressed as confidence regions in Θ. However, it is often of interest to make statements about the uncertainty of *individual* parameters, rather than *simultaneous* statements about multiple parameters together. In this context it is important to distinguish between *simultaneous* and *point-wise*, i.e., one-at-a-time, intervals. For a simultaneous CI, the uncertainty of the estimated parameters is stated for multiple parameters together, i.e., the computed confidence *region* captures the true parameters θ^* with *coverage probability* α Kullback [1939], Neyman [1937]. In comparison, a *point-wise* CI holds for that parameter alone, i.e., the *coverage probability* for capturing the single parameter is α.

To create *simultaneous scalar intervals* for each parameter, the higher dimension region can be *projected* onto each parameter, as illustrated in Fig. 2. Such projected simultaneous intervals should not be confused with *point-wise* CIs, nor should their *combined* confidence be stated as α Johnson and Wichern [2007], i.e., the coverage probability of all *projected* intervals holding is *not* α. The projected shadow of a higher order simultaneous confidence region is larger than the point-wise intervals Johnson and Wichern [2007], Press et al. [2007], as illustrated in Fig. 2.

Since Θ typically has more than two dimensions,

graphical presentation of confidence regions requires projections of some form. In such cases, care should be taken to clearly state the resulting confidence level. Just as for the elliptic region projected onto a single parameter axis in Fig. 2, a higher dimension ellipsoid projected onto a plane will give a larger elliptic *shadow* projection than a confidence ellipse computed for just two parameters in the plane.

2.2.6 Diagnosing identifiability by analysing uncertainty

Determining if a parameter is *structurally* or *practically* identifiable is important if the parameter value itself is of interest, i.e., if a *physical interpretation* of the parameter is assumed. A link between the uncertainty of a parameter, in the form of a likelihood based confidence interval as presented in Section 2.2.2, and the *structural* and *practical identifiability*, was given in Raue et al. [2009].

Structural non-identifiability is caused by *redundant* parametrisation of the model equations, such that a sub-set of the parameters θ_s has no effect on the observable outputs y, and is therefore independent of the experimental conditions $\mathcal{K}$ Raue et al. [2009]. Hence, there exists a manifold in the parameter space Θ where the objective function $\ell(\theta)$ has a constant value. Further, it is possible to obtain a functional relation between the parameters θ_s which describes this equipotential manifold in the objective function. Consequently, a likelihood based confidence interval will be unbounded in both direction, i.e., $[-\infty, +\infty]$, for the *structurally non-identifiable* parameters in θ_s Raue et al. [2009].

In contrast, *practical* non-identifiability is caused by a lack of dynamic information about the system in $\mathcal{K}$ and hence a direct result of the *experimental design* and data acquisition process. Unlike structural identifiability, practical identifiability is not clearly defined in the literature Raue et al. [2009]. However, an elegant definition is found in Raue et al. [2009], where practical non-identifiability is diagnosed if the corresponding likelihood based confidence region, as in Eq. (18), is extended to infinity in decreasing and/or increasing direction , i.e., the objective function stays below a specific threshold Δ_α in at least one direction, despite the presence of a well defined optimum $\hat{\theta}$. Observe that the use of *likelihood* based confidence regions is necessary for determination of practical non-identifiability, since the *asymptotic* CI will always be symmetric and also finite if $\Sigma_{i,i} > 0$, and therefore cannot present the necessary characteristics for diagnosing practical non-identifiability Raue et al. [2009].

In Raue et al. [2009] the definition of parameter identifiability is presented as a true/false question. For the case of structural identifiability, this is clearly appropriate. However, practical identifiability is a function of the system excitation during data acquisition, and hence the dynamic information content in $\mathcal{K}$. Therefore it may be appropriate to treat practical identifiability as a *quantity*, rather than a *true/false* property. If the computed confidence region for a parameter is *wide*, that parameter may arguably be considered *less* identifiable than a parameter with smaller confidence region. In particular, comparing parameters estimated from two different data-sets of the same system with different levels of excitation, the resulting CIs of the same parameter may have different widths. Hence, it is reasonable, and intuitively satisfactory, to relax the definition of practical non-identifiability given in Raue et al. [2009] to also include parameters with *abnormally wide* CIs. Unfortunately, relaxing the diagnostic criteria in this way leads to a *cognitive judgment* on what width of a CI is *abnormal* for any specific parameter. Resolving this question requires using system specific knowledge, and is further complicated by variations in scale of the parameters which makes normalisation a prerequisite for comparing CIs for different parameters.

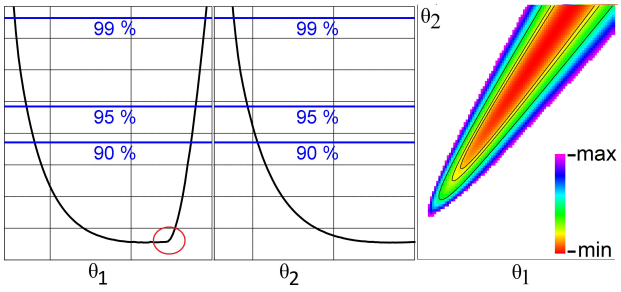

Figure 3: The log-likelihood profile of two inter-dependent parameters is plotted in the plane of both parameters (right panel), with corresponding projections to each of the two parameters.

2.2.7 Inter-dependent parameters and the effect of constraints on projections

When projecting a higher order region onto a single parameter θ_i or a plane $\Theta_{i,j}$, it is important to consider inter-dependent parameters. A projection of a higher dimensional region in Θ will, due to parameter inter-dependence, result in a projection that is wider than any cross-section, since the dependency information in the higher order structure is lost in the projection Johnson and Wichern [2007]. Further, if the parameter space Θ is constrained or bounded, inter-dependent parameters can introduce *artefacts* in the

projection of one parameter, caused by the constraints on another, inter-dependent, parameter.

An example of these phenomena is shown in Fig. 3. The *right* panel shows a log-likelihood profile of two parameters as a heat-map in the plane of both parameters. The *left* and *centre* panel show the same profile, projected onto each parameter axis. First, observe that the 2D profile in the right panel shows that the parameters are inter-dependent, since there is a clear linear relationship between the two parameters. Next, observe that the one-dimensional profiles, which are projections of the two-dimensional surface onto each parameter, are wider than any cross-section taken from the 2D profile. Finally, observe that for parameter θ_1 the profile contains a sharp *bend*, highlighted by a red circle in Fig. 3. When comparing to the full 2D profile, it is clear that this *bend* is actually an *artefact* in the θ_1 profile, introduced by the constraint on θ_2, i.e., $\theta_{2,\min} < \theta_2 < \theta_{2,\max}$, and the inter-dependence between the parameters.

2.3 Uncertainty estimation and analysis methods

In this section, a selection of methods for estimation of uncertainty and identifiability analysis is presented with some illustrative examples. More extensive examples of these methods are given in Section 3.

2.3.1 Uninformed Random Picking

It is often helpful to visualise the shape of the objective function g in the parameter space Θ. Initially, the optimal parameter estimate, the existence of a well defined optimum, and/or the number of optima, is typically unknown. Hence, a method which requires no assumptions about the objective function $g(\theta)$ and the parameter space Θ, is desirable. An intuitive approach is to evaluate the objective $g(\theta)$ for some selected *set* of parameters $\theta_{\{K\}} = \{\theta_k : k \in 1, ..., K\}$ and plot the resulting θ_k vs $g(\theta_k)$ as a *scatter plot* for each parameter. A simple way of selecting $\theta_{\{K\}}$ is by use of randomisation: drawing the parameters uniformly across Θ such that $\theta_k \sim \mathcal{U}(\theta_{min}, \theta_{max})$ for $k \in \{1, ..., K\}$. The resulting scatter plots will show that there exists an optimal *front* in Θ which corresponds to the projection of the objective $g(\theta)$ onto each parameter axis. Of course, a large number of the randomly drawn points in $\theta_{\{K\}}$ are not located near the optimal front. However, by randomised selection with *large* K, typically on the order of 10.000 to 500.000 or higher depending on the dimensionality of Θ, the plots will contain *enough* data near the optimal front to *visually* inspect the shape of the objective function. Subsequently, the existence of a well defined optimum, presence of flat regions, and the number of modes, can be ascertained. This method is often named *Uninformed Random Picking (URP)* Hoos and Stützle [2004], where the term *uninformed* reflects the fact that no prior assumptions is used in the choice of θ_k.

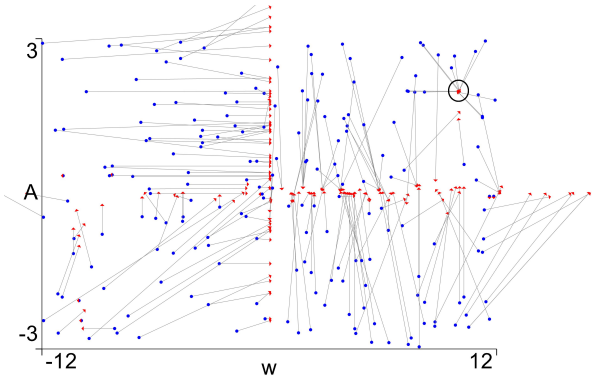

Figure 4: The Random initial guess method is used to test for local minima when estimating the amplitude (A) and frequency (w) of a simulated sine wave with added noise. Out of $K = 200$ repeated optimisations, only 12 correctly find the true parameters θ^*(marked by a black circle in the figure). Blue dots mark the randomised initial guess and red dots mark the parameters obtained after optimisation.

The resulting plots will be a projection of K parameter vectors onto each of the n_θ parameter axis in Θ. Hence, the method suffers from the challenges related to such projections, as discussed in Section 2.2.7. Examples of the use of the URP method is given in Section (3.1.1).

2.3.2 Randomised initial guess

A variation of the *Uninformed Random Picking* method from Section 2.3.1 is to use randomisation to uniformly draw the initial guess θ^0 and subsequently optimise all parameters, i.e., repeatedly solve the optimisation problem in Eq. (1) K times, with a randomised initial guess $\theta^0_k \sim \mathcal{U}(\theta_{\min,}, \theta_{\max})$ for $k \in \{1, ..., K\}$. This method, although simple, can be a good test to check the convergence of the parameter estimation method. If repeated executions of the optimisation algorithm $\mathcal{A}$ returns significantly different optimal estimates $\hat{\theta}_k$ depending on the initial guess θ^0_k, a *physical interpretation* of the estimated parameters as constants given by the system $\mathcal{S}$ must be considered *questionable*. This may indicate a problem with parameter identifiability, which should be analysed further. Since the optimisation algorithm $\mathcal{A}$ allows *directed* exploration

of the objective function, the number of iterations K can be much lower than required for the URP method, say, 10 to 500. As for the URP method, the choice of K depends on the dimensionality of the parameter space Θ.

Additionally, since optimisation is performed from a number of different starting points in Θ, this method can be useful to identify local minima in the parameter space, provided the number of iterations K is large enough to cover the parameter space with reasonable density.

The resulting optimal estimates $\hat{\theta}_k$ are plotted together with the initial guesses θ_k^0 to indicate the trajectories of the optimisation algorithm $\mathcal{A}$. The results can either be plotted for two parameters against each other, forming a projection of the corresponding optimisation trajectories onto a plane of two parameters, or they can be plotted for each parameter; θ_k^0, $\hat{\theta}_k$ vs $g\left(\theta_k^0\right)$, $g\left(\hat{\theta}_k\right)$. The resulting plots will give a good visualisation of the projected shape of the objective function.

An example of this method is shown in Fig. 4. By simulating a sine wave $y(t) = A\sin(wt)$, where the parameters are amplitude A= 2 and frequency $w = 10$, and adding Gaussian noise of standard deviation 0.5, a calibration data-set is created. When estimating the parameters of this simple model, a large number of local optima are found, especially for $A = 0$ or $w = 0$. Hence, the estimated solution $\hat{\theta}$ is highly dependent on the initial guess θ^0. Only 12 of the $K = 200$ repeated optimisations correctly obtain $\hat{\theta} = \theta^*$. This example shows the importance of considering local minima when estimating parameters. It is also interesting to observe that there are different patterns of trajectories in each of the four quadrants of the plot. These variations are caused by the optimisation method $\mathcal{A}$, and shows that also the chosen algorithm for optimisation can have a strong influence on the parameter estimate.

2.3.3 Hessian of the likelihood function

A commonly used method for estimating the uncertainty of the estimated parameters is to utilize the shape of the objective function $g(\theta)$ directly by calculating the curvature around the optimal estimate $\hat{\theta}$, by computing the *Hessian* of $\ell(\theta)$; $H = \nabla^T \nabla \ell(\theta)\big|_{\theta=\hat{\theta}}$. The covariance of the estimated parameters can be computed as $\Sigma_\theta = 2H^{-1}$, where the factor 2 is included to compensate for previously dropping the factor $\frac{1}{2}$ in the definition of $\ell(\theta)$ in Eq. (10). The elements of H are approximated as Kristensen et al. [2004], Raue et al. [2009]:

$$h_{i,j} \approx \left(\frac{\partial^2}{\partial\theta_i \partial\theta_j}\ell(\theta)\right)\Bigg|_{\theta=\hat{\theta}} \tag{19}$$

which can be numerically computed using, e.g., *central difference approximation*. Observe that the Hessian is by definition *symmetric*, which is a drawback if the *shape* of the objective function is *non-symmetric* around the optimum. Observe also that, from general optimisation theory, while the curvature of any objective function g could be considered an approximation of uncertainty Nocedal and Wright [2006], the estimation of parameter covariance Σ_θ from the Hessian requires that a log likelihood objective $g(\theta) = \ell(\theta)$ is used. This method obtains directly the parameter covariance Σ_θ which can be used to construct an asymptotic confidence region as in Eq. (14).

2.3.4 Profile likelihood

As discussed in Section 2.2.2, *likelihood based* CIs are often considered superior to *asymptotic* CIs Raue et al. [2009]. Further, parameter *distributions* are arguably a more descriptive representation of uncertainty than confidence regions. Hence, obtaining parameter distributions based on the likelihood function is an attractive tool for parameter analysis. An elegant method for computing such distributions is the *profile likelihood* (PL) method presented in Deconinck and Roels [2017], Murphy and Van der Vaart [2000], Raue et al. [2009]. The PL method explores the parameter space by optimising the parameters in *two* steps, rather than simultaneously as in Eq. (1). The *profile likelihood* $\ell_{\mathrm{PL}}(\theta_i)$ is defined as the minimum log likelihood for a given θ_i when the remaining parameters are freely optimised Raue et al. [2009]:

$$\ell_{\mathrm{PL}}(\theta_i) = \min_{\theta_{j\neq i}} \ell(\theta_{j\neq i}; \mathcal{M}, \mathcal{K}, \theta_i) \tag{20}$$

Values of θ_i must be chosen prior to optimising the remaining $\theta_{j\neq i}$ Raue et al. [2009]. A straightforward solution, if the objective function g is well behaved within the constraints of Θ, is to use a *brute force* approach with an even sampling of θ_i. Alternatively, a two-sided gradient decent algorithm, using a freely optimized parameter vector as a starting point, can be applied Maiwald and Timmer [2008], Raue et al. [2009]. The resulting likelihood distribution can be plotted as a function of θ_i and subsequently analysed according to the definitions of structural and practical identifiability for *likelihood based confidence intervals* Deconinck and Roels [2017], as discussed in Section 2.2.2. A threshold can be applied to the constructed profile, as described in Section 2.2.2, where, by *Wilks'*

Theorem Wilks [1938], the threshold Δ_α can be drawn from the $\chi^2_{\alpha,n_{\text{df}}}$ distribution. The freely estimated $\hat{\theta}$ has n_θ degrees of freedom (d.o.f.), while the PL estimate has $n_\theta - 1$ d.o.f., hence the threshold Δ_α is computed with $n_{\text{df}} = 1$.

Observe that since the PL method essentially projects the n_θ dimensional space Θ onto the single parameter θ_i, by freely estimating the remaining parameters, the PL method tends to overestimate the width of the likelihood based confidence interval if parameters are not independent, as discussed in Section 2.2.7

2.3.5 Two-dimensional profile likelihood

In order to improve the PL methods projections under the influence of inter-dependent parameters, the method can be modified to hold out *two* parameters rather than one, i.e.:

$$\ell_{\text{PL2}}(\theta_i, \theta_j) = \min_{\theta_{k\neq i,j}} \ell(\theta_{k\neq i,j}; \mathcal{M}, \mathcal{K}, \theta_i, \theta_j) \tag{21}$$

This projects the parameter space Θ onto the plane of θ_i and θ_j; $\Theta_{i,j}$, which results in a two-dimensional distribution that can be analysed in a similar way to the one-dimensional PL Raue et al. [2009], using the definition in Eq. (18) and discussed in Section 2.2. The PL2 results can be plotted as topological surfaces Raue et al. [2009], which can be used to diagnose parameter *inter-dependence*, since the two-dimensional projections are capable of representing *relationships* between parameters. These projections constitute an exhaustive search over the plane $\Theta_{i,j}$. Hence, both local and global optima can be obtained from inspection of the projected profiles.

Applying a confidence threshold to the PL2 method produces *confidence regions* in the $\Theta_{i,j}$ plane. Based on confidence thresholds computed from the χ^2-distribution, a similar interpretation of these two-dimensional topologies can be applied to diagnose identifiability by requiring that the region is bounded in all directions Raue et al. [2009]. If there is an unbounded equipotential *valley* with a *constant optimal* log likelihood, the parameter is structurally non-identifiable. If the interval or region is unbounded in some direction but still has a well defined optimum, this indicates a practically non-identifiable parameter Raue et al. [2009]. Observe that since $\hat{\theta}$ has n_θ free parameters while the PL2 estimate has $n_\theta - 2$, this gives $n_{\text{df}} = 2$ for the computation of Δ_α from the χ^2-distribution in Eq. (18).

While the extension of the PL method to create projections in the plane $\Theta_{i,j}$ is intuitive, and the resulting plots exhibit some interesting characteristics as tools for analysing parameter identifiability and inter-dependence, this modification strongly increases the computation time of the method. To create the projections of $\ell(\theta)$ onto $\Theta_{i,j}$, a large number of objective function evaluations must be performed. Using a brute force sampling of $\Theta_{i,j}$ with N steps for each parameter returns N^2 pairs of parameter values, each of which requires optimisation of the remaining parameters; $\theta_{k\neq i,j}$. This process must be repeated for each combination of parameters, which further increases the computational burden. Hence, the method requires careful use of parallelisation and software engineering to be computationally feasible. Of particular importance is utilising the fact that neighbouring points in $\Theta_{i,j}$ are likely to have similar optimal values for $\theta_{k\neq i,j}$. Hence, using previously optimised free parameters as a *warm-start* for computing new $\ell_{\text{PL2}}(\theta_i, \theta_j)$ points significantly reduces computation time.

Due to the extensive computation time for this method, it is advisable to initially perform exploratory analysis with relatively low number of steps N, with subsequent higher resolution analysis in specific regions of interest. However, the initial analysis must use a discretisation resolution sufficiently detailed to find the regions of interest. The number of resolution steps for each parameter which is required for successful application of the PL2 method depends on the problem, and should be found by experimentation.

Finally, observe that when a brute force discretisation of $\Theta_{i,j}$ is used, the resulting set of optimised parameters constitutes an exhaustive search of the discretised parameter space Θ. Hence, an estimate $\hat{\theta}$, which is *globally optimal* within the accuracy and bounds allowed by the brute force discretised Θ, can be obtained by taking the minimum from all the $\ell_{\text{PL2}}(\theta_i, \theta_j)$ profiles.

2.3.6 Bootstrapping for dynamic models

The data samples of a time-series are not independent. Hence, the traditional bootstrapping method of randomly drawing individual samples with replacement is not applicable, because the sample to sample dependency information would be lost in the generated pseudo data-set Kunsch [1989], Politis [2003]. A popular modification of the bootstrapping method is to divide the original data into blocks, either overlapping or non-overlapping, with uniform or randomly chosen lengths and/or starting points Politis [2003]. In this paper, two versions of block based bootstrapping for time series data is considered; *non-overlapping block bootstrap* Lodhi and Gilbert [2011] and *stationary bootstrap* Politis and Romano [1994]. The difference between these two approaches is in how the data is separated into blocks. Each method is outlined below.

The idea behind all bootstrap methods is to generate multiple pseudo data-sets, in order to estimate the

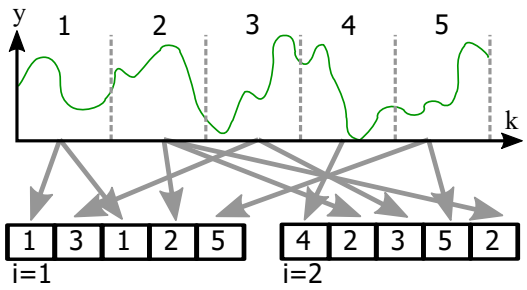

Figure 5: An example of simple block bootstrapping with $K = 5$ blocks, creating $M = 2$ pseudo data-sets.

variance of some estimate, e.g., dynamic model parameters. Hence, the bootstrapping procedure must be repeated M times, such that each iteration produces a *different* pseudo data-set, and hence a *different* parameter estimate $\hat{\theta}_j$. Note that any objective function could potentially be combined with this type of bootstrapping, e.g., the ballistic SSE approach in Section 2.1.3 or the likelihood $\ell(\theta)$ in Section 2.1.2.

Based on these M estimated parameter vectors, the *mean* parameter estimate, and the *covariance* of that mean estimate can be computed, i.e:

$$\hat{\theta} = \frac{1}{M}\sum_{j=1}^{M}\hat{\theta}_j$$
$$\Sigma_\theta = \frac{1}{M-1}\sum_{j=1}^{M}\left(\hat{\theta}_j - \hat{\theta}\right)^2 \tag{22}$$

Confidence regions on the form of Eq. 14 can then be constructed for the *mean parameter estimate*, where the threshold Δ is drawn from the *F-distribution* Johnson and Wichern [2007].

Additionally, the resulting M parameter estimates can be plotted as scatter plots or as histograms, either for individual parameters or for combinations of two parameters. Observe that these plots suffer from the same limitations related to the projection of high dimensional parameter space Θ onto single parameter axis as discussed in Section 2.2.7.

The first bootstrap method, *non-overlapping block bootstrapping*, is achieved by dividing the data-set $y_{[N]}$ and $u_{[N]}$ into K blocks of length l. Let $y_{[l]}^{(i)}$ and $u_{[l]}^{(i)}$ be block i of measured system outputs and inputs, respectively, where $i \in \{1, ..., K\}$. Each block is constructed by taking a consecutive sequence of samples from the original data, $y_{[N]}$ and $u_{[N]}$, starting from

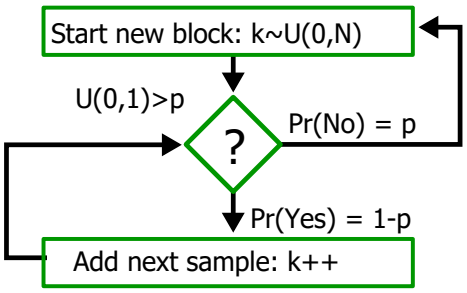

Figure 6: Simplified block diagram of Stationary Bootstrapping.

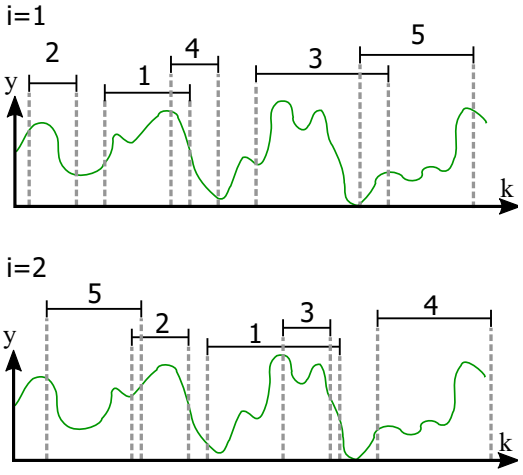

Figure 7: Illustrative example of two iterations of stationary bootstrapping, resulting in five blocks of data, possibly overlapping, with uniformly drawn starting point and geometric length distribution.

sample $l \cdot (i-1)$, such that:

$$y_{[l]}^{(i)} = \left[y_{l\cdot(i-1)+k} \;:\; k \in [1, ..., l]\right] \tag{23}$$
$$u_{[l]}^{(i)} = \left[u_{l\cdot(i-1)+k} \;:\; k \in [1, ..., l]\right] \tag{24}$$

A pseudo time-series data-set is then created by drawing K blocks randomly with replacement, as illustrated in Fig. 5. As with traditional bootstrapping, some blocks will not be drawn, while others may be drawn multiple times. Hence, some data points will not appear in the new pseudo data-set, while other data points will appear multiple times. This is shown in Fig. 5, where for the first pseudo data-set block 1 appears twice, as the 1st and 3rd block, while block 4 is not included.

An alternate method for drawing random blocks of data is the *stationary bootstrapping* method Politis and Romano [1994], where blocks of data are constructed with a random length of geometric distribution Politis [2003]. The advantage of this approach is to create

bootstrapped data-sets that are themself stationary series Politis [2003], Politis and Romano [1994].

The stationary bootstrap method is realised by use of a probability test, and consists of two steps. First, the starting point of each block is drawn uniformly across all N original samples, $k_0 \sim \mathcal{U}(0, N)$. Next, with probability $1-p$, add the next consecutive sample from the original series, or with probability p, start a new block by again uniformly drawing a new starting point. This test on p is repeated until the combined length of all blocks is approximately N. The resulting blocks length will follow a geometric distribution such that the probability of block i having length m is $P_r(l_i = m) = (1-p)^{m-1} p$. This process is illustrated in Fig. 6. The expected length of each block is $E(l) = \frac{1}{p}$ and the expected total number of blocks is $E(K) = \frac{N}{p}$. An illustrative example of two iterations of stationary bootstrapping is shown in Fig. 7. Comparing Fig. 7 to the non-overlapping block bootstrapping in Fig. 5 shows the difference in the two methods, in that the first method has non-overlapping blocks of uniform length, which is randomly recombined to create the pseudo data-set, while the stationary bootstrap method uses randomisation to choose both the start and length of each block.

Since both these approaches, indeed, all block based bootstrapping methods for time-series data, involve dividing the original time-series data into blocks and recombining them to form new pseudo data-sets, the question of how to join together multiple randomly selected blocks into a new complete data-set arises Kunsch [1989]. For estimation of parameters for dynamic system models using a data-set that is essentially segmented non-consecutive blocks, an intuitive solution is to compute the objective function $g(\theta)$ for each block and aggregate the results. If the objective is defined on summation form as in Eq. (6), the overall objective function for a block segmented data-set of K blocks can be defined:

$$g_{\mathrm{B}}\left(\theta; \mathcal{M}, y_{[N]}, u_{[N]}\right) = \sum_{i=1}^{K} g^{(i)}\left(\theta; \mathcal{M}, y^{(i)}_{[l_i]}, u^{(i)}_{[l_i]}\right)$$

The initial conditions for evaluating the objective for each block, $g^{(i)}$, such as the initial state, must be determined for each block, rather than for the whole data-set as in Eq. (1). If the states are measurable, the choice of initial state for each block can be obtained from the measurements. Alternatively, the initial state can be treated as an unknown parameter and estimated for each randomly drawn block.

An important consideration when performing block based bootstrapping on time series data for dynamic model parameter estimation, is the *consistency* of the dynamic information in the data. If certain segments of the data contain significantly *less* dynamic information than the rest, e.g., if the system is in *steady state* for parts of the original data-set, some iterations of the bootstrap procedure may return pseudo data-sets that are *less informative* w.r.t. parameter estimation. These pseudo data-sets may produce *practically non-identifiable* parameters Raue et al. [2009], which manifest as *outliers* among the M bootstrap estimates. Such outliers will significantly effect the computed covariance in Eq. (22). Hence, it is important to consider the *consistency* of dynamic information in the original data, prior to applying bootstrapping methods.

2.3.7 Consistency of dynamic information in calibration data

An intuitive method for testing the *consistency* of dynamic information content in data is to draw a set of overlapping, consecutive, data segments, taken equidistant across the data-set. Each segment is of length l, and extracted from starting points $w \cdot (i-1)$, where w is the step length;

$$y^{(i)}_{[l]} = \left[y_{w\cdot(i-1)+k} \,:\, k \in [1, ..., l]\right] \tag{25}$$

$$u^{(i)}_{[l]} = \left[u_{w\cdot(i-1)+k} \,:\, k \in [1, ..., l]\right] \tag{26}$$

The approach constitutes a *moving window* that travels across the data-set with step length w. The segment length l and the step length w are considered tuning parameters and should be determined experimentally. For each segment, a parameter vector $\hat{\theta}^{(i)}$ is estimated by minimising the objective $g^{(i)}\left(\theta; \mathcal{M}, y^{(i)}_{[l]}, u^{(i)}_{[l]}\right)$. Note that this is fundamentally different from the bootstrapping approach, since no randomisation is used to combine multiple segments and the parameter estimation is performed separately for each consecutive segment. As for the block based bootstrapping methods in Section 2.3.6, the initial conditions needed to evaluate $g^{(i)}$ must be obtained for each segment, either as estimated parameters or directly from observations if the states are measurable.

For *each* segment, some appropriate method of uncertainty estimation, e.g., the *Hessian* method of Section 2.3.3 or the *Profile Likelihood* method in Section 2.3.4, is used to evaluate the *uncertainty* and/or *identifiability* of the estimated parameters. By plotting the results as a function of the segment starting point $w \cdot (i-1)$, and observing how the parameter uncertainty and/or identifiability changes with time as the window is moved, the consistency of dynamic information in the data can be evaluated. Observe also that if the PL method is used, the results should be plotted as the relative log likelihood $\ell(\theta) - \ell\left(\hat{\theta}_i\right)$, since the optimal log likelihood will be different for each segment.

If parameter calibration from different segments of the data produce significantly different uncertainty estimates, this indicates an inconsistency in dynamic information, which subsequently can influence uncertainty estimation methods based on block bootstrapping. Observe that since a small subset of the calibration data is used, the uncertainty estimates for each segment will be larger than what is obtained using the complete original data-set.

In addition to test the consistency of dynamic information by estimating the *uncertainty* for each step, the method also produces an estimate of the optimal *parameters* $\hat{\theta}^{(i)}$ for each segment. These estimates can be used to test if the optimal model parameters change over time for a specific data-set. If the parameters are interpreted as constants of the physical system, time variation of θ can be an indication of unsatisfactory calibration data. Arguably, *unexpected* time variation of parameters may also indicate an oversimplified model structure, such that the calibrated parameters are affected by unmodelled time-varying disturbances, resulting in variations in the parameter estimates over time.

2.4 Summary

Section 2 of this paper has presented the theoretical foundation for a number of methods that can be used to analyse the parameter estimation problem for dynamic models, in particular the *identifiability* and *uncertainty* of the estimated parameter. Which method is best suited for a particular application largely depends on the application, and what type of analysis is of interest. The aim of these methods is to obtain *accurate* dynamic system models, but also to *validate* the estimated parameters in the presence of *aleatoric* and *epistemic* uncertainty. In many applications the choices for experimental design is limited. Hence, parameters must be estimated under less than ideal conditions. It is especially important in these cases to carefully analyse the resulting parameter estimates in the context of *identifiability* and *uncertainty*. In engineering applications, parameters are often assumed, quite reasonably from a detailed physical understanding of the underlying system, to be constants of the physical system. However, due to the effects of measurement noise, unmodelled disturbances, insufficient dynamic information, modelling errors and simplifications, etc., it may not be possible to obtain an unambiguous *estimate* of the parameters. Hence, the methods presented in this section can be valuable engineering tools for providing a thorough analysis of the parameter estimation problem. In the sequel, examples of the application of these methods to two experimental cases are presented. These examples illustrate the kind of insight that can be gained by applying the methods to practical parameter estimation problems.

3 Experimental cases

In this section, the methods presented in Section 2 are demonstrated on two test cases. The first case is a simple first order model with a single input. The parameters of the model are calibrated using data obtained by simulating the same model, with added, randomly generated, measurement noise. Hence, there is no *epistemic* uncertainty in the parameter estimation, only the *aleatoric* uncertainty of the output measurement noise. The second case is an example of a grey-box model, specifically a thermal network model of a building, which aims to predict temperature variations. These models are particularly interesting from a parameter estimation and analysis perspective, since they are constructed *cognitively* based on *naive* physics, and hence have significant *epistemic* uncertainty in them.

3.1 First order dynamic model

A first order model with input is defined as:

$$\dot{x} = -ax + bu \tag{27}$$

$$y = x + v_k \tag{28}$$

where $v_k \sim \mathcal{N}(0, \mathcal{V})$ is the Gaussian distributed measurement noise, u is the model input and the parameters are $\theta = [a,\, b]$. By Laplace transformation, the transfer function from input to output is obtained as:

$$H(s) = \frac{y(s)}{u(s)} = \frac{b}{s+a} = \frac{K}{\tau s + 1}, \tag{29}$$

which is a low-pass filter with exogenous input, with gain $K = \frac{b}{a}$ and time constant $\tau = \frac{1}{a}$. The output y is hence simply the low-pass filtered input u plus the measurement noise.

The model is excited with six different input signals, each a total of 10 second of data at $\Delta t = 0.01$, presented in Fig. 8. The six data-sets are chosen to demonstrate the effect of different types of excitation on the various methods to be tested. As shown in Fig. 8, the first three data-sets are a step (STP), a square wave (SQR), and a sine wave (SIN), where short-hand names are given to simplify tabulating results in the sequel. The square and sine wave have a signal period $T = 2s$. The remaining three excitation signals are *pseudo random binary sequences* (PRBS), i.e. signals generated by drawing a sequence of *random* binary numbers and transforming those into a non-uniform square wave signal. The length in time of each bit, i.e., bit-length, is $0.1s$ (PR1), $0.2s$ (PR2) and $0.5s$ (PR5) for the last

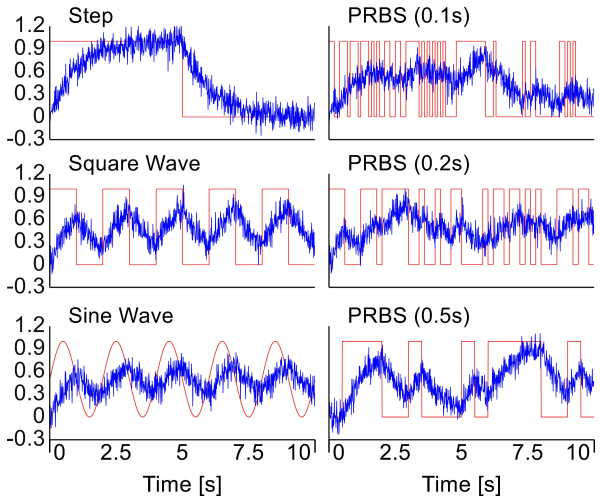

Figure 8: Six different input signals (red) are used to excite the first test case model. For each signal, the model is simulated deterministically to obtain the output (blue). The three left panels show the Step, Square wave and Sine wave signals. The Square wave and Sine wave both have a period of $T = 2s$. The right panels show the three pseudo random binary sequence (PRBS) signals, which differ in what length in time each bit in the sequence represents ($0.1s$, $0.2s$ or $0.5s$).

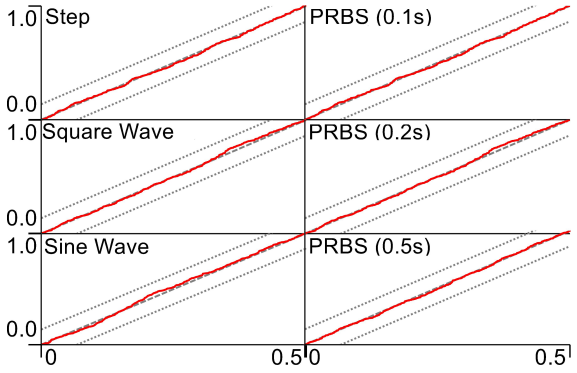

Figure 9: Cumulative Periodograms for all six data-sets show that the residuals are close to Gaussian distributed, well within the 95% confidence bounds.

three data-sets. Hence, a value of true/false in the PRBS indicates input $u = 1/u = 0$ for $0.1s$, $0.2s$ or $0.5s$, respectively.

The model is simulated for each of the six input signals, with parameters $a = 1$, $b = 1$ and added measurement noise $v_k \sim \mathcal{N}\left(0, 0.1^2\right)$, to obtain an output. Hence, for this model, the true parameter vector θ^* is known. The resulting input-output data-sets are used as $y_{[N]}$ and $u_{[N]}$ in the following tests. Since there is no diffusion term in the state transition in Eq. (27), and the calibration data is simulated with the same model for which parameter analysis is performed, the model is exact, hence $\mathcal{W} = 0$. As expected, due to the simplicity of the model, and the simulated data-set, the residuals are close to Gaussian, as shown by the CP diagrams in Fig. 9

3.1.1 Uninformed Random Picking and Profile likelihood

When starting to analyse the parameter space, a good first step is to visualise the shape of the objective function $g(\theta)$ in the parameter space Θ. Usually, the existence of a well defined, *unambiguous*, optimum $\hat{\theta}$ is not known initially. Hence, a good starting method is the *Uninformed Random Picking (URP)* method described in Section 2.3.1. The result of using URP as an exploratory tool on the first order model is shown in Fig. 10. Additionally, the *Profile Likelihood* (PL1) method is applied to show that both methods obtain the same optimal *front* across Θ. Observe from Fig. 10 that the grey dots correspond to each of $K = 50.000$ randomly drawn θ_j, each simulated to compute $g(\theta_j)$, while the red line is the PL1 profile. The PL1 profile corresponds closely to the optimal front obtained by the URP.

Plotting the results together with the likelihood profile, shows that the same information, the shape of the objective, is obtained by both methods. Hence, it is interesting to compare the methods on computation time and implementation. For this simple model, the execution time of URP ($K = 50.000$) and PL1(500 steps in θ_i) are $12s$ vs $17s$, hence the computation time is short enough to be insignificant. However, for larger models, there may be significant differences. The PL1 method requires optimisation of $n_\theta - 1$ parameters for each step in θ_i, hence, a large number of parameters significantly increases the load on the optimisation algorithm. In contrast, the URP method requires no optimisation, but is affected by the dimensionality of Θ due to the dispersion of the randomly drawn points. With large number of parameters, K must be chosen large enough that the randomly drawn parameters reasonably covers the whole space Θ, which results in longer computation time.

An interesting observation when comparing PL1 and URP, since they both give essentially the same result, is that URP is significantly easier to implement, since it does not require an optimisation algorithm. For some applications, this may be a distinct advantage.

Next, observe that both the URP and the PL1 method rely on projections to plot the results as functions of a single parameter. These projections are known to overestimate the width of the pro-

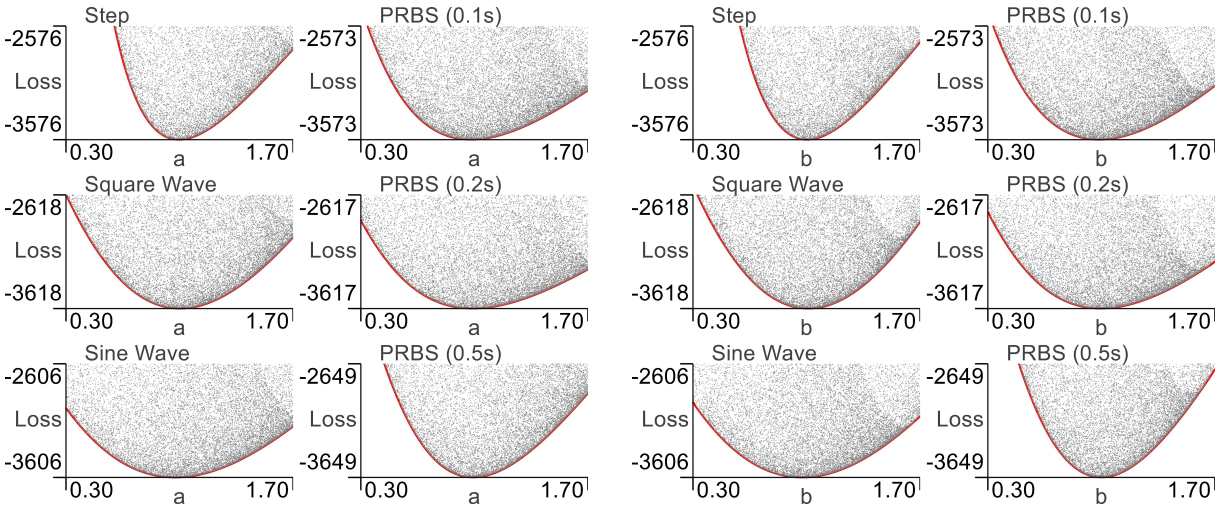

Figure 10: Exploratory analysis of the $\ell(\theta)$ objective using the PL1 (red) and the URP (grey) methods. Results for parameter a (left) and b (right) show that both parameters are unambiguously identifiable for all six data-sets.

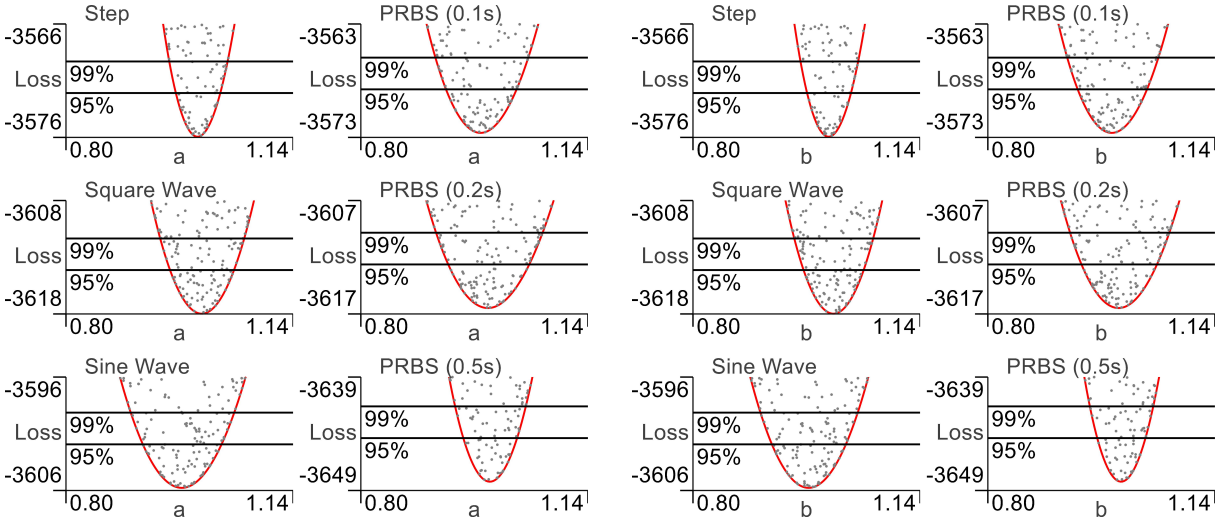

Figure 11: PL1 and URP results, zoomed in around the optimum, for parameter a (left) and b (right).

files/intervals, as discussed in Section 2.2.7.

Finally, observe that the plots in Fig. 10 are obtained as a form of exploratory analysis, hence with wide bounds on Θ and subsequently with a large range on the objective score axis; relative log likelihood $\ell(\theta) - \ell\left(\hat{\theta}\right) < 1000$ is used here. These plots are interesting as a first step, but for the purpose of estimating uncertainty of the optimal estimate $\hat{\theta}$, only the immediate neighbourhood of $\hat{\theta}$ is of interest. Hence, Fig. 11 shows the same results but with different scaling on the axis. Here, the width of Θ is significantly reduced, and also the range in objective score is reduced to a more reasonable 10. This likelihood range allows for adding confidence thresholds at α equal to 90% and 95%. From Fig. 11 it is immediately apparent that the Step and PRBS (0.5s) data-sets produce narrower shapes around $\hat{\theta}$ than the other four data-sets, which indicates better estimation accuracy, i.e., tighter confidence bounds from the applied thresholds.

3.1.2 Randomised initial conditions

Another useful method, especially as an initial exploratory analysis tool, is the use of randomised initial conditions with subsequent optimisation, discussed in Sec. 2.3.2. The result of applying this method is shown in Fig. 12, where the results are plotted as a vs b, i.e., both parameters against each other. Hence, Fig. 12 shows the whole parameter space Θ for this model. As shown, the optimum $(1, 1)$ is obtained for

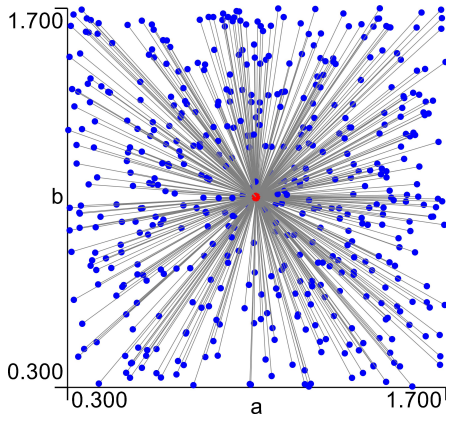

Figure 12: Randomised initial conditions shows that the optimum is globally unambiguous in Θ and obtained independent of the initial guess θ_0. Results are shown only for data-set Step (other five sets show the same behaviour).

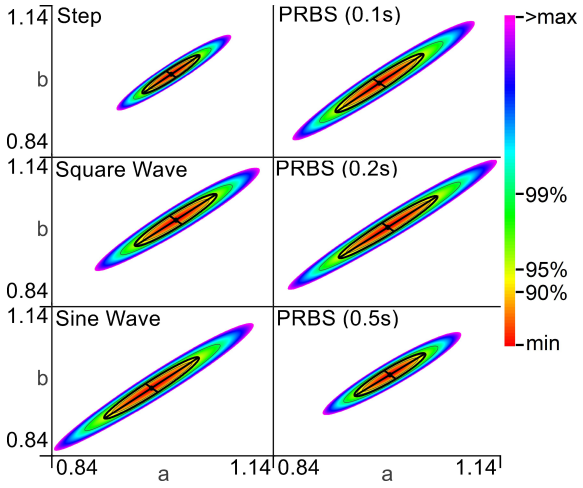

Figure 13: PL2 heat-maps with 90%, 95% and 99% confidence iso-lines, for all six data-sets, with added 95% confidence ellipses (thick line), computed from the Hessian of $\ell(\theta)$, for comparison.

all $K = 500$ randomly drawn initial guesses, which shows that the optimum is unambiguously obtainable in Θ, and not influenced by the initial guess θ_j^0. For comparison, see Section 2.3.2 where another example of this method is shown in Fig. 4 in which there are a large number of local minima. Together with the results in Section 3.1.1, Fig. 12 shows that the objective function has a well defined single global optimum. The major difference between results from the six different data-sets is the shape of the objective around the optimum, and subsequently the accuracy of the obtained parameter estimate, which will be further discussed in the sequel.

3.1.3 Profile Likelihood 2D and Hessian

A natural next step is to perform a detailed analysis of the neighbourhood around $\hat{\theta}$, i.e., the parameter ranges obtained from the PL1 analysis shown in Fig. 11. To analyse the parameter space, the two-dimensional Profile Likelihood (PL2) method from Section 2.3.5 is used. The results, shown in Fig. 13, use the same range for all six data-sets in order to directly compare the obtained profiles. For comparison, ellipses as in Eq. (14), computed by using the Hessian method from Section 2.3.3, are added to the PL2 plots. Observe first from Fig. 13 that the parameter distributions in Θ are very well approximated by the Hessian based ellipses. This is expected, due to the simplicity of the model and the simulated data with added Gaussian noise. Both methods use the same objective function $\ell(\theta)$, with the only difference being that the Hessian method assumes a quadratic distribution to compute elliptic regions.

Table 1: Standard deviations of parameters computed with the Hessian method.

Data	a	b	σ_a	σ_b	$\sqrt{\sigma_a\sigma_b}$
STP	0.997	1.003	0.016	0.016	0.015
SQR	1.003	1.010	0.022	0.021	0.021
SIN	0.974	0.973	0.026	0.025	0.025
PR1	0.980	0.986	0.024	0.024	0.023
PR2	0.991	0.996	0.027	0.025	0.025
PR5	0.993	1.000	0.019	0.017	0.018

Next, observe that the elliptic confidence regions in Fig. 13 are rotated at an approximately 45 degree angle, or equivalently from Table 1 that the covariance $\sigma_a\sigma_b$ between the two parameters is significant, compared to the variance of each variable. This indicates that the parameters are *dependent*, which is expected from Eq. (29), since $K = \frac{b}{a}$. The parametrisation of the model in Eq. (27) was chosen specifically to demonstrate this point. Subsequently, as discussed in Section 2.2.7, the PL1 projections from Fig. 11 are too wide. Indeed, by projecting the PL2 results in Fig. 13 onto each of the two parameter axes, the resulting profiles would be exactly the results from the PL1 method. Hence, it can be observed that the PL1 method significantly over-estimates the width of the parameter profiles due to parameter inter-dependence. Note that it is recommended to attempt resolving parameter inter-dependence by choosing a different parametrisation in Eq. (27), e.g. choosing the parameters K and τ such that the state transition equation becomes $\dot{x} = \frac{1}{\tau}(-x + Ku)$.

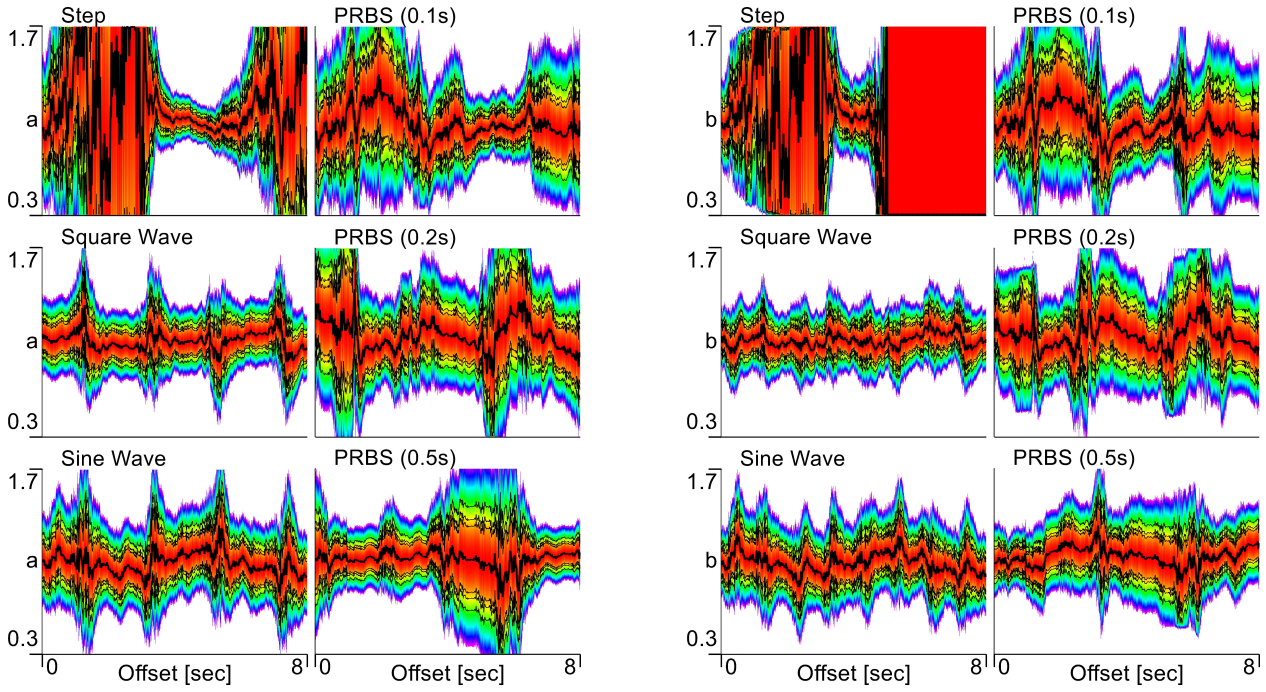

Figure 14: Consistency of dynamic information for identification parameters a (left) and b (right) is examined using a window length of $l = 200$ samples, equivalently 2 seconds. The window offset is varied on the interval $t_0 \in [0, 8[$ seconds in steps of $w = 1$, or equivalently $\Delta t = 0.01$. At each step, the Profile Likelihood method is used to evaluate identifiability of the parameters using the information in the window. The results are plotted as a heat-map with time offset on the horizontal axis, and the thick line represents the optimal parameter estimate for each window. The parameters are examined on the interval $[0.3, 1.7]$.

Finally, observe from inspection of Fig. 13 and the corresponding quantified standard deviations and covariance of the parameters in Table 1, that the Step and PRBS (0.5s) data-sets provide slightly more accurate estimates of the parameters, compared with the other four. The variations in parameter estimation uncertainty, and correspondingly the shape of the neighbourhood around $\hat{\theta}$ in Θ, are caused by the use of different excitation signals. Hence, the differences between the results illustrate the well known fact that the choice of excitation signal during experiments directly influences the parameter estimation uncertainty.

3.1.4 Consistency of dynamic information

Next, it is of interest to assess the consistency of dynamic information in the data-sets, using the Moving Window method described in Section 2.3.7. The results of applying the Profile Likelihood (PL), described in Section 2.3.4, to data segments of length l taken equidistantly across the original data with step length w, is shown in Fig. 14. The PL method provides better estimates of the uncertainty and *identifiability* for the data in each step of the moving window, compared with the Hessian, since it can represent *asymmetric* distributions. The results, which for this method is a function of parameter θ_i and the time offset $w \cdot (i-1)$, are plotted as heat-maps with confidence iso-lines at 90%, 95% and 99%. Figure 14 shows that there is a significant difference between the Step data-set and the other five sets in that the Step data-set has large segments where the parameter uncertainty is high, i.e., large equipotential bands in the parameter direction. This indicates that there is insufficient dynamic information in these segments of the data to obtain good parameter estimates. Observe also that for the Square and Sine wave data-sets, the results are the least affected by the window offset, hence, these data-sets contain the most consistent dynamic information. Similarly, the optimal estimate, marked by a thick black line in Fig. 14, is showing significant fluctuations for the Step data-set, while for the Square and Sine Wave data-sets, the estimates are mostly consistent w.r.t. the time window offset.

These considerations will be especially important in the sequel, when block based bootstrapping methods are used, but the results are also interesting in themselves, as a way to test the dynamic information con-

Table 2: Bootstrap results ($M = 200$ iterations), Case A: Simple ($K = 10$), Case B: Simple ($K = 5$), Case C: Stationary ($p = 0.005$).

Data	#	a	b	σ_a	σ_b	$\sqrt{\sigma_a \sigma_b}$
STP	A	0.958	0.985	0.080	0.073	0.074
	B	1.009	1.018	0.063	0.058	0.060
	C	1.013	1.015	0.065	0.070	0.061
SQR	A	1.049	1.033	0.026	0.015	0.019
	B	1.005	1.025	0.014	0.007	0.008
	C	1.001	1.014	0.018	0.016	0.015
SIN	A	0.975	0.965	0.017	0.025	0.015
	B	0.970	0.970	0.018	0.021	0.016
	C	0.977	0.977	0.028	0.025	0.026
PR1	A	0.976	1.006	0.068	0.058	0.059
	B	0.979	0.983	0.024	0.028	0.025
	C	0.949	0.951	0.033	0.034	0.032
PR2	A	1.063	1.040	0.051	0.039	0.043
	B	1.026	1.032	0.031	0.018	0.023
	C	0.999	1.007	0.029	0.026	0.027
PR5	A	0.991	0.997	0.046	0.040	0.038
	B	0.994	1.011	0.025	0.036	0.030
	C	1.014	1.027	0.020	0.023	0.020

tent of different excitation signals, especially when using calibration data obtained from physical systems with limited choices in the experimental design.

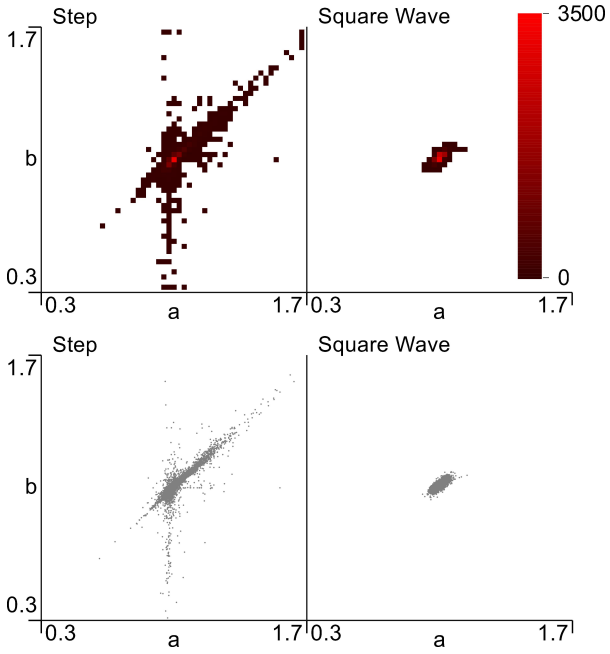

Figure 15: Histogram (top) and scatter plot (bottom) showing how data with poor dynamic information content (left panels, Step data-set) induces outliers in the results. Plots are obtained using Stationary bootstrap with tuning parameter probability $p = 0.005$ and $M = 10000$ iterations.

3.1.5 Bootstrapping

The use of bootstrapping methods for dynamic data to estimate the uncertainty of estimated parameters, is discussed in Section 2.3.6. Here, the simple block bootstrap, with block lengths $l = 100$ and $l = 200$, respectively 1 and 2 seconds of data, is tested and compared with the Stationary Bootstrap method using $p = 0.005$. The results, after $M = 200$ iterations, are presented in Table 2. First, observe that, as expected based on the results in Section 3.1.4, the Step data-set shows considerably higher covariance than the other data-sets. Since large segments of the Step data-set does not contain sufficient dynamic information for parameter estimation, some of the randomised pseudo-data-sets created by bootstrapping does not contain enough information to estimate parameters, hence the higher covariance. This is further illustrated by Fig. 15, which shows how the Step data-set produces significantly larger spread in parameter estimates, compared with the Square Wave data-set. Note that a much higher number of iterations, $M = 10.000$, was used for Fig. 15 in order to obtain a good histogram illustration. From Section 3.1.4, the Square and Sine Wave data-sets are known to have significantly better consistency of dynamic information across the data-set compared with the Step data-set. Hence, more consistent parameter estimates with lower variance is obtained from the block-based bootstrapping methods.

Next, observe that the Stationary Bootstrap method produces approximately the same results as the block based bootstrapping with $l = 200$, in this case. This may be explained by the Stationary Bootstrap using $p = 0.005$ which gives an expected block length also of 200.

Finally, observe that for the datasets for which the dynamic information is of sufficient consistency, the estimates of the parameter uncertainty Σ_θ for cases B and C in Table 2 are similar to those obtained from the Hessian method in Table 1.

3.1.6 Frequency information in input and output

A commonly used method of examining dynamic information content in data is to compute a frequency spectrum using the Fast Fourier Transform (FFT) algorithm. Due to its widespread use and popularity, computationally efficient implementations exist, which makes this an easily accessible tool for analysing data. Applying FFT to the six data-sets, both the input signal and the measured output with noise, gives the res-

ults shown in Fig. 16. Comparing the FFT results to those obtained by the parameter analysis methods presented previously provides some interesting insight into the *differences* in results obtained from each of the six excitation signals.

First, comparing the Sine and Square wave data-sets, observe that the Sine wave has only one frequency component at $f = 0.5\,Hz$, excluding the mean signal level component at $0\,Hz$, while the Square Wave contains also higher order harmonics of the base frequency. However, since the model is essentially a low-pass filter with a critical frequency of $\frac{1}{2\pi}$, these higher order harmonics are damped, thus having only limited effect on the model output. Note however that despite having almost identical frequency information in the output y, the spectra for the input u differ significantly. The fact that these higher order harmonics in the input spectra are damped, thus not significantly infuencing the output y, is also informative w.r.t. the input-output relationship of the model, thus producing slightly smaller confidence regions for the Square Wave data-set.

A similar observation can be made from comparing the three PRBS data-sets. For the sets with bit-length $0.1s$ and $0.2s$ more of the input signal power is located in the damped frequency region of the model. Hence, the PRBS signal with bit-length $0.5s$ produces better parameter identification results, since more of the frequency information is passed through the model.

Finally, comparing the Step and Square Wave data-sets shows why the Step data-set produces the narrowest confidence regions from parameter estimation. Observe that the Step data-set contains the most signal strength in the frequency pass-band of the model. Since more of the information in the input u affects the output y, the parameter estimation methods produce estimates with lower uncertainty.

3.1.7 Comparing excitation signals

An interesting observation can be made from comparing the results of the various methods for all six data-sets. While the Step data-set gives the *best* estimation accuracy for the Likelihood based methods, such as the Hessian curvature method and the Profile Likelihood method, it also produces the *worst* estimation accuracy when *bootstrapping* is used. The reason for this can be observed from the consistency plot in Fig. 14. While the Step data-set contains segments of data that are largely *uninformative* for the purpose of estimating dynamic model parameters, the segments that *do* contain sufficient information produce the highest accuracy estimates. The PL confidence bands produced when the step change in the input is included in the moving window are the *tightest* among all the results produced, hence, give the lowest estimation uncertainty. However, since bootstrapping randomly selects segments of data, there will be some bootstrapped pseudo data-sets that do not contain data from the informative segment of the Step data-set and therefore produce outlier parameter estimates such as the ones shown in Fig. 15.

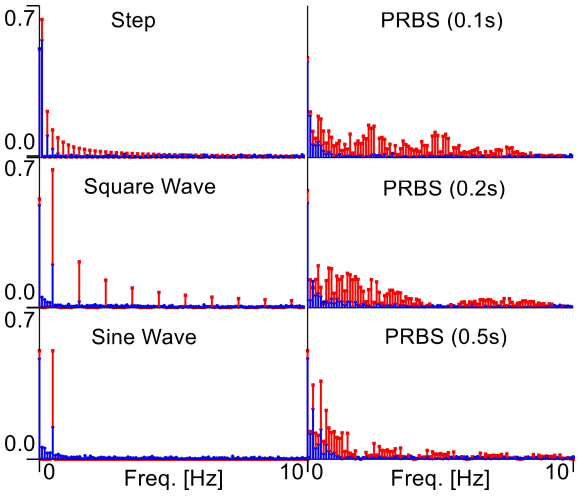

Figure 16: The Fast Fourier Transform (FFT) can be used to obtain a frequency domain representation of the dynamic information content in both input and simulated output. Input u (red) and output, including measurement noise, y (blue) is shown.

This example shows that assessing dynamic information content for model calibration is not straightforward, even in this simple case. Hence, it is useful to apply a method for evaluating the consistency of dynamic information, such as the one presented in Section 2.3.7.

3.1.8 Computation time

An important consideration for any numerical estimation method is the computation time it takes to obtain results. Typically, computation time depends on hyper parameters of the method, such as the resolution of likelihood profiles or the number of iterations for bootstrapping and randomisation based methods. Further, computation time also depends on the dimensionality n_θ of the parameter space Θ. Some selected examples of computation times for the previously presented results are given in Table 3. Since all the parameter estimation methods are based on a large number of simulations of a known model structure with varying parameters θ and a set of measurement data $y_{[N]}$ and $u_{[N]}$, the computation time is also influenced by the complexity of the model, the choice of Kalman Filter implementation for computation of residuals and the size N of the calibration data-set. Further, since

Table 3: Computation time for the LP model from the Step data-set. The other data-sets produce comparable execution times.

Method	Time
PL1 (resolution 500)	$\sim 13s$
PL1 (resolution 5000)	$\sim 90s$
URP ($K = 50.000$)	$\sim 19s$
URP ($K = 500.000$)	$\sim 160s$
Rand. Initial Conditions ($K = 50$)	$\sim 7s$
Rand. Initial Conditions ($K = 500$)	$\sim 57s$
Moving Wnd. (res. 200, $w = 1$, $l = 200$)	$\sim 575s$
Moving Wnd. (res. 200, $w = 1$, $l = 100$)	$\sim 317s$
Moving Wnd. (res. 100, $w = 1$, $l = 200$)	$\sim 311s$
Moving Wnd. (res. 200, $w = 10$, $l = 200$)	$\sim 57s$
PL2 (400×400 resolution)	$\sim 2800s$
Hessian	$\sim 1s$
Bootstrap A ($M = 200$)	$\sim 14s$
Bootstrap B($M = 200$)	$\sim 23s$
Bootstrap C($M = 200$)	$\sim 24s$
Bootstrap C($M = 10.000$)	$\sim 990s$

most of the presented methods use numerical optimisation, computation times can be influenced by optimisation related effects, such as variations in computation time due to obtaining estimates at different local minima. Hence, it is interesting to compare and discuss the computation time for a known model and data-set.

First, observe that for the PL1 and randomisation based methods, i.e., URP and Random Initial Conditions, the computation time is approximately linear in the resolution/randomised iterations. However, the number of iterations of the randomisation methods required to adequately explore the parameter space Θ depends on the dimensionality. The PL1 method however projects the likelihood function of the parameter space onto each parameter axis. Hence, the effect of high dimension parameter spaces will be more significant for the randomisation based methods than the PL1 method.

Next, comparing the PL2 exhaustive search of Θ with the ellipsoid approximation obtained from computing the Hessian, the difference in computation time is around three orders of magnitude. Further, the PL2 method projects the likelihood function of the entire parameter space Θ onto each possible plane. Hence, the computational time increases exponentially with the number of parameters n_θ.

The computation time of the Moving Window analysis, which applies the PL1 method on a moving window sub-set of the data, is shown to be approximately linear in the step length w. This is expected since the step length directly determines for how many windows of data the PL1 method is executed. Further, the computation time is also linear in the window length l. The number of window data sub-sets is $n_\mathrm{w} = \frac{N-l}{w}$ which is only somewhat affected by l. However, the PL1 method is approximately linear in the length of the data used, which results in the computation time for the Moving Window analysis being also approximately linear in l. Finally, the computation time is linear in the PL1 *resolution*, which was previously shown for the PL1 method applied to the full data-set. Naturally, the same applies when the method is used on a small sub-set of the data.

Finally, the computation time is approximately the same for the bootstrap methods in cases B and C. This is expected, since Case B uses blocks of fixed length $l = 200$ while the Stationary Bootstrap method in Case C uses $p = 0.005$, which gives the average block length $E(l) = \frac{1}{p} = 200$. Comparing this to Bootstrap Case A shows that both bootstrap methods are approximately linear in the expected block length. Additionally, since bootstrapping must be repeated M times in order to simulate running M experiments, the computation time is also approximately linear in M.

3.2 Thermal network model of a building

The second test case consists of a thermal network model of a small experimental building located at the Porsgrunn Campus of the University of South-Eastern Norway (USN). Thermal network models are created *cognitively* based on *naive* physical descriptions of the thermodynamics of the buildings, and can be expressed as Resistor-Capacitor (RC) circuit *analog* models Berthou et al. [2014], Deconinck and Roels [2017], Fux et al. [2014], Madsen and Holst [1995]. Specifically, the R3C2 model, partially based on the R4C2 model presented in Berthou et al. [2014], is created by ignoring heat convection and radiation. Due to the strong simplification used in these models, they contain significant *epistemic* uncertainty, in addition to the *aleatoric* measurement uncertainty induced by acquiring data from a physical building. Due to the simplified nature of the model, the assumption $\mathcal{S} \in \mathcal{M}(\Theta)$ is clearly unjustified here. However, it may still be possible to obtain $\hat{\theta}$ such that $\mathcal{M}\left(\hat{\theta}\right)$ is a good approximation of $\mathcal{S}$. Hence, it is interesting to analyse the parameter space Θ of this model to evaluate the identifiability and estimation uncertainty of $\hat{\theta}$.

The model circuit equivalent is shown in Fig. 17. The model has two outputs: the room temperature T_b and the wall surface temperature T_w, and two inputs: the consumed power by an electric heating element $\dot{Q}$ and the outside temperature T_∞. Five components form the model structure: the thermal res-

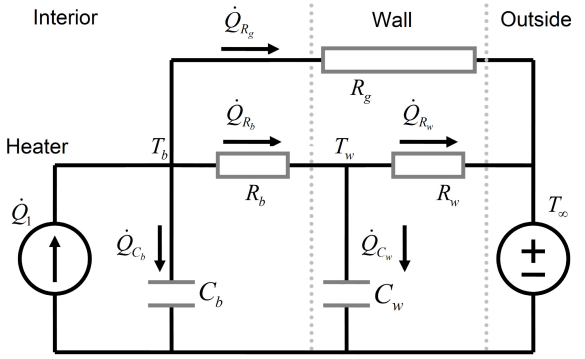

Figure 17: Simplified thermal network model with three resistors and two capacitors.

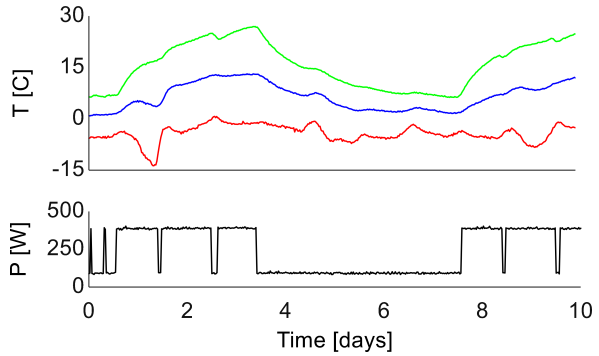

Figure 18: Calibration data for the R3C2 model. Temperatures T_∞ (red), T_w (blue) and T_b (green), and also the power consumption Q, was recorded in February 2018.

istance between room air and wall R_b, the building envelope R_w, and the thermal resistance of windows and doors R_g. The two capacitances C_b and C_w represent the thermal capacitance of the building interior and envelope, respectively. Additionally, the process and measurement noise covariances W and V are also estimated as model parameters, since they are needed in the Kalman filter. Both covariance matrices are assumed diagonal, adding a total of four noise related parameters to the vector θ.

A calibration data-set for this model is shown in Fig. 18. The data was collected from the experimental building during February 2018, using a pre-installed data acquisition system and set of sensors Brastein et al. [2018].

3.2.1 Profile Likelihood of R3C2 model

Initially, both PL1 and PL2 methods were used to perform an exploratory analysis of the parameter space of the R3C2 model. The results of these analyses, presented in Figs. 19 and 20, show that there is a problem with the parameter space of this model, particularly that the parameter R_b and R_w are inter-dependent. Observe from Fig. 20 that the R_b vs. R_w plot shows a linear relationship. Hence, the PL1 results for these two parameters in Fig. 19, which can be considered a projection of the PL2 result onto the individual parameter axis, show a large equipotential flat region which extends up to at least $5\frac{\mathrm{K}}{\mathrm{W}}$. Observe also that for R_b the PL1 profile makes a sharp bend at around $4.5\frac{\mathrm{K}}{\mathrm{W}}$, such that the profile is bounded for R_b. However, as discussed in Section 2.2.7, inter-dependent parameters can cause *artefacts* in the PL plots, due to the bounds on one parameter having a limiting effect on the other dependent parameters. The bend in the profile of R_b is an example of such an *artefact*.

Subsequently, the R3C2 model is found to be over-parameterised. After some experimentation, based on previous experience with the model Brastein et al. [2019b], the resistor R_g is removed from the circuit model in Fig. 17, in an attempt to make the remaining parameters identifiable. The resulting model, named R2C2, is used in the sequel and further analysed.

3.2.2 Profile Likelihood of R2C2

The first analysis performed on the reduced R2C2 model is a combination of the PL1 method and the URP method. The results, presented in Fig. 21, show that all four parameters are now identifiable, since the likelihood based confidence intervals are bounded with a clearly defined minima. Secondly, comparing URP to PL1 shows that although the URP method successfully captures the *general shape* of the objective function around $\hat{\theta}$ using $K = 500.000$ randomly drawn parameters, it is not enough to properly capture the optimal front. Hence, there is some small difference between the PL1 and URP results. By its use of numerical optimisation, the PL1 method successfully finds the optimal profile in likelihood space for each parameter. The main result from the application of PL1 is to obtain reasonable bounds θ_min and θ_max on Θ for further analysis, something for which the PL1 method is ideally suited.

Next, the PL2 method is used to further analyse the parameter space Θ, in particular to test for inter-dependency of parameters and further study the identifiability. For comparison, the Hessian method from Section 2.3.3 is used to compute the covariance of the estimated parameters Σ_θ, and subsequently compute an elliptic confidence region for the true parameters θ^*. The Hessian ellipses are superimposed on the PL2 heat-maps in Fig. 22. Two interesting observations can be made from these results. First, the results show that after removing R_g, all parameters are identifiable, i.e., the confidence regions are bounded, given the

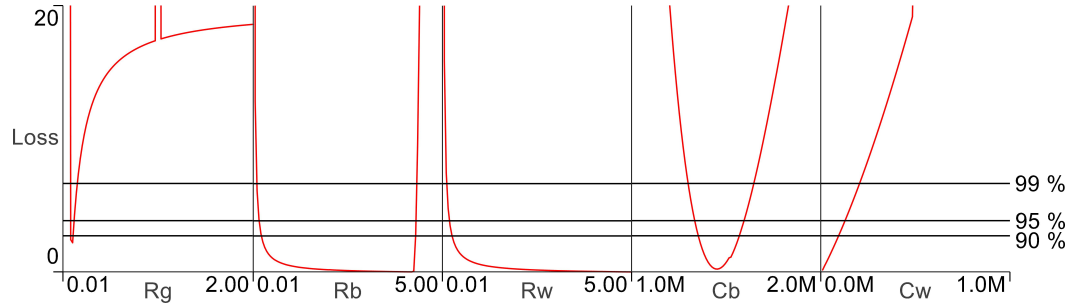

Figure 19: PL1 results for the R3C2 model.

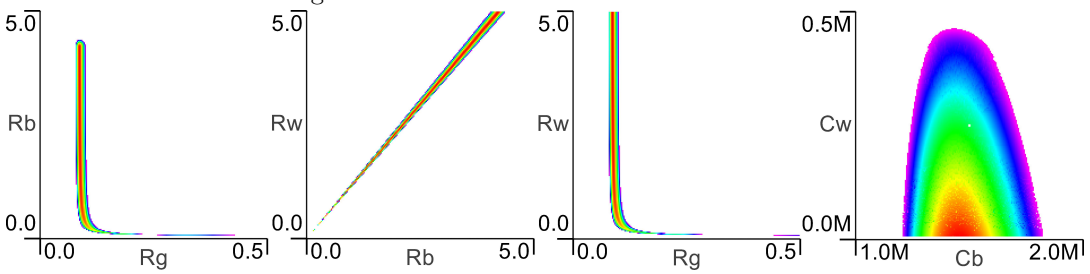

Figure 20: Selected PL2 results for the R3C2 model shows that the parameters R_w and R_b are inter-dependent.

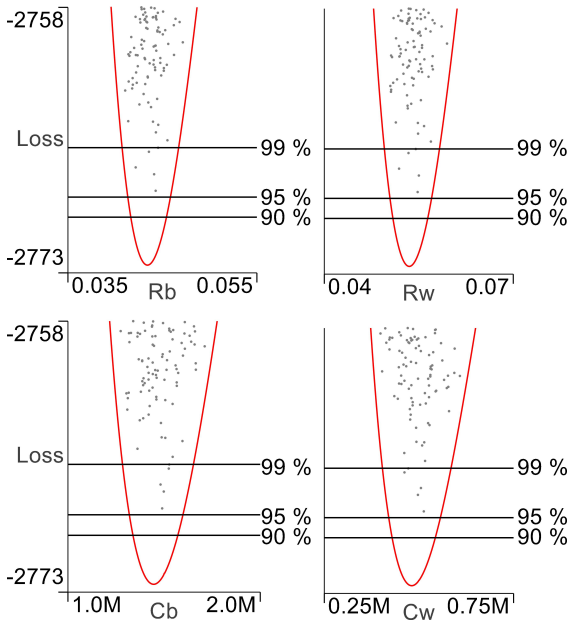

Figure 21: PL1 and URP results for the reduced R2C2 model show that even with $K = 500.000$ randomly drawn parameter vectors, the coverage is not good enough, since the optimal front from the PL method is not the same as that of the URP method. However, the shape of the objective is still approximated by the URP method, indicating its usefulness also for higher dimensional parameter spaces.

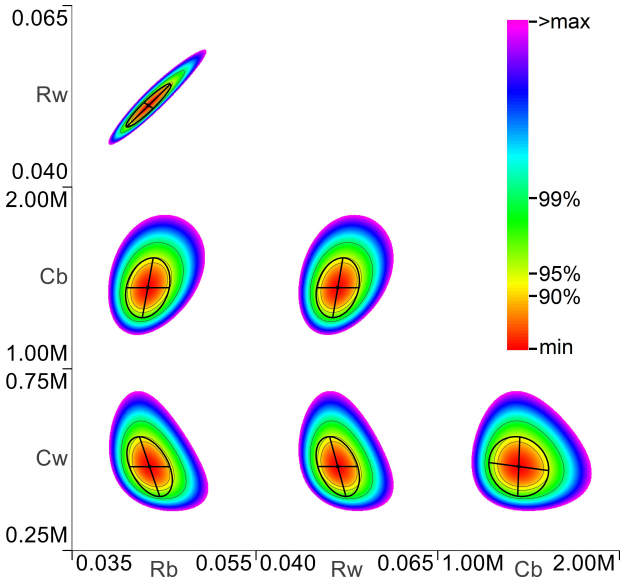

Figure 22: PL2 and Hessian ellipses (thick black) for the R2C2 model. Iso-lines trace the 90%, 95% and 99% confidence bounds computed from the PL2 results, based on the $\chi^2_{n_\mathrm{df}}$-distribution with $n_\mathrm{df} = 2$. The Hessian method is used to compute Σ_θ and superimpose an elliptic approximate confidence region at $\alpha = 95\%$.

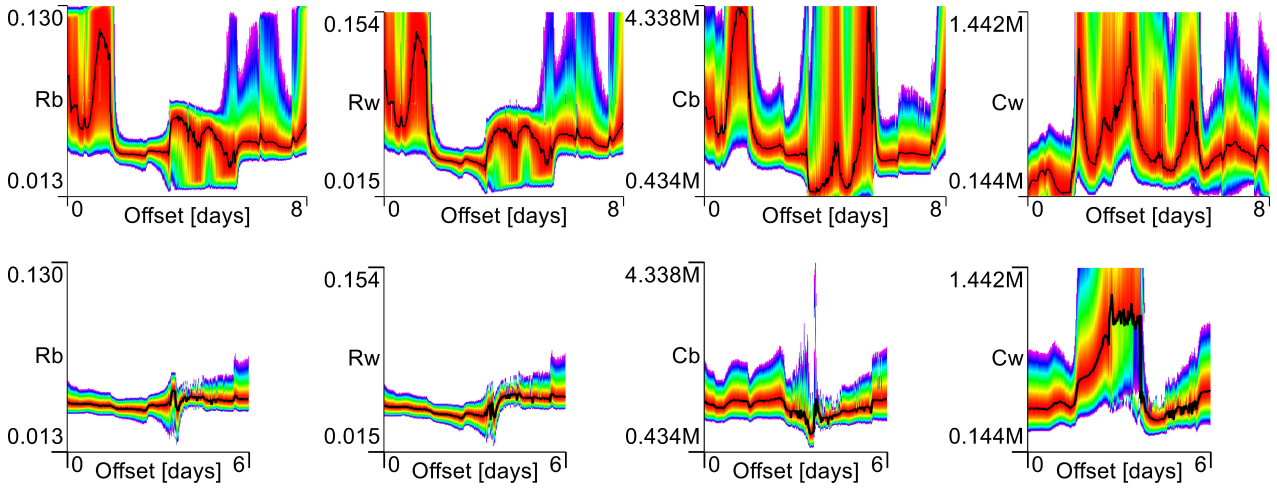

Figure 23: Consistency of dynamic information for the R2C2 model. The top four plots show the results of applying PL1 to a moving window of length 96 (2 days), while the lower four plots use a window length of 192 (4 days).

data in Fig. 18. The parameters R_b and R_w are still inter-dependent, as illustrated by the rotation of the approximately elliptic PL2 profile, but there is still a clearly defined optimum. Second, the Hessian method produces a reasonable approximation of the 95% isoline confidence bounds in all the projected parameter planes. Where the PL2 method and Hessian method differ, it can be observed from Fig. 22 that the PL2 method, which by brute force computation captures the true projection of $\ell(\theta)$ onto $\Theta_{i,j}$, finds profiles that are not quite elliptic. The discrepancies observed visually therefore seem reasonable w.r.t. the shape of the PL2 profile. Observe for example that the C_b vs C_w profile is elongated in the increasing direction of both parameters, hence the discrepancy between PL2 and Hessian ellipse is mostly located towards the decreasing parameter directions.

Table 4: Optimal parameters with normalised standard deviations computed with the Hessian method for the R2C2 model.

	R_b	R_w	C_b	C_w
$\hat{\theta}_i$	0.0434	0.0512	1.446×10^6	0.481×10^6
$\frac{\sigma_i}{\hat{\theta}_i}$	0.0233	0.0210	0.0467	0.0702

The optimal parameters, which are the same for both PL2 and Hessian methods, are shown in Table 4 together with the standard deviations computed from inverting the Hessian, normalised over the optimal parameters.

3.2.3 Consistency of dynamic information

Since it is of interest to test bootstrapping methods also on the R2C2 model, a verification of the dynamic information content is first needed. A typical challenge for building thermal behaviour models is the restrictions on experimental design, since weather, including outside temperature, is a model input. Additionally, there are limitations to acceptable ranges of indoor temperature and limited available input power for heating, which further complicates the experimental design for this type of models. Therefore, model calibration must often be performed on *low informative* data. Hence, methods that can evaluate the quality of the dynamic information in the data is of interest. By using the PL1 method for a moving window of data, as discussed in Section 2.3.7, it is possible to obtain a visual diagnosis of estimation accuracy and parameter identifiability for segments of the data.

As shown in Fig. 23, the estimation accuracy of parameters in a window of length 96 samples (2 days) is somewhat poor for significant segments of data, in particular for the first part of the data-set. The parameter C_w is particularly difficult to identify, even for the 192 sample (4 days) window. From inspecting the calibration data in Fig. 18 this result is expected, due to the limited variation observed in temperature T_w. For the parameters R_b, R_w and C_b, the consistency test shows that the uncertainty is mostly consistent in time, with only minor variations, for the 4 day window case. The results also show that the optimal value for these three parameters do not vary significantly over time, for the window length of 4 days. However, the parameter C_w is estimated with significant time vari-

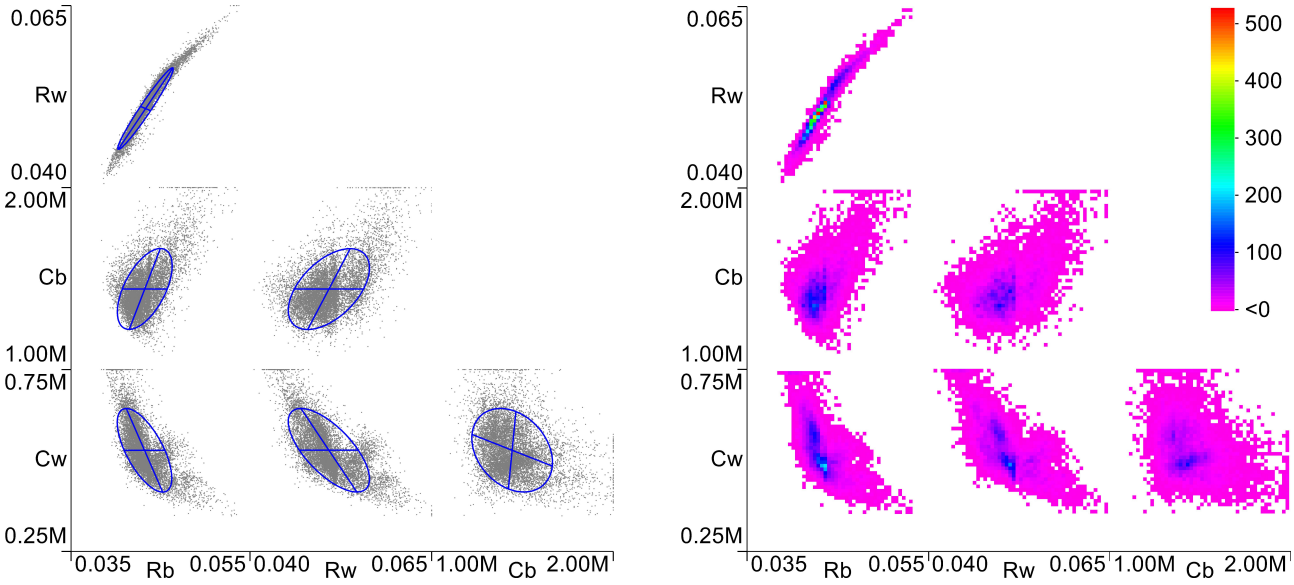

Figure 24: Bootstrap results for the R2C2 model, with $M = 10.000$ iterations represented as scatter plots with 95% confidence ellipses, simultaneous for the projected parameters, for the mean estimate (left) and corresponding 2D histograms (right). Both the scatter plots and the histograms are presented as two-dimensional projections onto each possible parameter combination plane $\Theta_{i,j}$.

ations also for the longest window length, as shown in the lower right panel of Fig. 23. This indicates poor identifiability of C_{w}, and may result in unsatisfactory results if block-based bootstrapping methods are used to estimate uncertainty.

3.2.4 Bootstrapping

The parameters of the R2C2 model is next analysed using Stationary bootstrapping with $p = 0.005$, which gives expected, i.e., average, block length $E(l) = 200$, since a window length of 192 samples (4 days) appears to be an acceptable choice based on the results in Fig. 23. The resulting mean parameters and normalised standard deviations, after running bootstrapping for $M = 10.000$ iterations, is shown in Table 5.

Table 5: Optimal parameters with normalised standard deviations computed with the Bootstrapping method for the R2C2 model.

	R_{b}	R_{w}	C_{b}	C_{w}
θ_i	0.0432	0.0509	1.443×10^6	0.528×10^6
$\frac{\sigma_i}{\theta}$	0.043	0.067	0.093	0.131

Comparing the results in Table 5 with Table 4, the estimated *mean* of the M bootstrapped iterations agrees well with the result obtained by optimisation and PL2 brute force exhaustive search. The normalised covariances obtained by bootstrapping, i.e., the covariance of M iterations of repeated generation of pseudo data-sets with subsequent parameter estimation, are approximately two times larger than those obtained by inverting the Hessian of the likelihood function. Considering the significantly different theoretical foundation of these two methods of uncertainty estimation, a difference of a factor of two or three may be considered a reasonable agreement between the two methods, in particular since the consistency test in Fig. 23 showed that the calibration data contains some low informative regions which can cause outliers in the bootstrapped parameter estimates. A histogram over all M iterations is shown in Fig. 24. Since the parameter space is of a dimension higher than two, the histograms are plotted as projections onto parameter planes $\Theta_{i,j}$, similar to the projected profiles obtained from the PL2 method. Interestingly, the shape of the histograms is similar to the PL2 profiles obtained in Fig. 22. However, due to the effect of outliers caused by some of the randomised pseudo data-sets being significantly less informative than the full data-set, the spread of the histogram. i.e., the covariance of the mean estimate, is larger than the covariance obtained from the Hessian in Table 4. Observe also the *clustering* of parameter estimates at the *edges* of the histogram plots, which indicates that for certain iterations of the bootstrap methods, the obtained parameters are located at the constraints of the parameter space Θ. This is a further indication that some pseudo data-sets are non-informative w.r.t. parameter estimation, since

the resulting parameters at the bounds of Θ deviate significantly from those obtained when the full data-set is used.

3.2.5 Computation time

Computation times for the various analysis methods applied to the R2C2 model are shown in Table 6. First, observe that the computation times are considerably longer than those found for the simple first order model in Table 3, e.g., the URP method with $K = 500.000$ randomly drawn parameters was completed in $\sim 160s$ for the first order model but took $\sim 0.15h = 540s$ for the R2C2 model. Despite using a dataset with only approximately half the number of samples, 480 vs 1000, the computation time for the R2C2 model is approximately 3.4 times longer. This extended computation time is caused by increased model complexity. First, the model has two states rather than one. Additionally, the R2C2 model uses a UKF rather than a standard KF, which further increases computational time. When analysing the R2C2 model, the software evaluates the model's equations ~ 540.000 times per second for a total of $\sim$ 1100 simulations per second. Comparably, the simpler first order model's equations are evaluated $\sim$ 3.100.000 times per second, for a total of $\sim$ 3100 simulations per second. Since the URP method does not use optimisation, model complexity, length of the data-set and the number of URP iterations K are the main factors that influence computation time, hence the results can be directly compared. Accounting for differences in data-set length, the increased model complexity of the R2C2 model, including its use of UKF with Runge-Kutta 4th order discretisation Runge [1895] of the state equation, increases computation time by approximately $\frac{540}{160}\frac{1000}{480} = 7$ times.

Next, observe that the stationary bootstrap, which shows similar results to the PL2 method, is about 40 times faster. This increased computation speed is obtained at the cost of inducing outlier estimates, caused by Bootstrapped pseudo data-sets that are less informative w.r.t. parameter estimation than the full data-set. Hence, due to these outliers, the uncertainty estimate is somewhat inflated compared to that obtained when computing the Hessian of the Likelihood function over the whole data-set.

Finally, observe from Table 6 that the Moving Window analysis computation time is only approximately linear in the window length l. The analysis using a longer window length of 192 is finished with a 1.62 times longer computation time, compared with the window length of 96. While this method is theoretically linear in window length l, the shorter window is less informative w.r.t. parameter estimation, as Fig. 23 shows. Hence, the task of the numerical optimiser is more challenging, which increases the computation time slightly for the shorter window. This example illustrates that calculating computation time for complex analysis methods is not straight forward. The Moving Window with PL1 method consists of both a numerical optimisation method, a Kalman filter implementation, the model structure, and the Profile Likelihood algorithm, all of which influence the computation time.

Table 6: Computation time for the LP model from the Step data-set. The other data-sets produce comparable execution times.

Method	Time
PL1 (resolution 500)	$\sim 4.35h$
URP ($K = 500.000$)	$\sim 0.15h$
Moving Wnd. (res. 200, $w = 1$, $l = 96$)	$\sim 3.60h$
Moving Wnd. (res. 200, $w = 1$, $l = 192$)	$\sim 5.77h$
PL2 (400×400 resolution)	$\sim 15.18h$
Stationary Bootstrap ($M = 10.000$)	$\sim 0.35h$

3.3 Method recommendations

Each of the methods presented in this paper has its advantages and disadvantages. Since they each compute and represent the uncertainty of estimated parameters in different ways, they can be used for different applications.

First, with regards to representation of uncertainty as profiles or regions, this is a question of usage. As an uncertainty estimate for comparison, regions or intervals may be preferable, since they can be quantitatively compared. Profiles are more descriptive, since they can represent how the uncertainty is *distributed* across an entire parameter domain. Hence, for applications where the parameters themselves are of interest, i.e., assumed to be determined by the physical properties of the system, representing parameters as distributions is perhaps preferable since they capture the most information about the underlying physical system.

Second, with respect to choosing what methods to use, the first question to consider is whether it is reasonable to assume that the parameters are well approximated by a Gaussian distribution, such that a quadratic approximation can be used to obtain ellipsoid regions for describing the uncertainty. In such cases, and when confidence regions rather than profiles are desirable representations, the Hessian method for computation of estimation covariance is preferable, due to its computational simplicity and speed. The Hessian method is based on analysing the curvature of the likelihood function $\ell(\theta)$ around an optimal estimate $\hat{\theta}$, which must first be obtained by calibration of all para-

meters and hence is subject to local minima problems. Hence, the Hessian method may only estimate the uncertainty of a pre-determined, *presumed* optimal, $\hat{\theta}$. Therefore, it should be ascertained, if possible, whether a particular $\hat{\theta}$ is a global or local optimum.

The *Profile Likelihood* (PL1) method Maiwald and Timmer [2008], Meeker and Escobar [1995], Murphy and Van der Vaart [2000], Raue et al. [2009], Venzon and Moolgavkar [1988] is an attractive choice if the practical identifiability of parameters is questionable. This method, unlike the Hessian based method, can represent non-symmetric confidence regions which can be used to diagnose identifiability Raue et al. [2009]. Further, the method allows representation as profiles, which may also be an advantage in some cases. The PL1 method can also be used for obtaining reasonable limitations on parameters in an exploratory analysis. Although it is known to give projections onto single parameters, which can be too wide if there are inter-dependent parameters, it is still a useful analysis tool.

The *Uninformed Random Picking* (URP) method Hoos and Stützle [2004] is a simple alternative to PL1, and provides approximately the same results if the number of randomly drawn parameters K is large enough. However, being a stochastic method, the distribution of randomly drawn parameters across parameter space can not be guaranteed. Hence, the optimal front in parameter space may not be detected unless a sufficiently large number of parameters is used. This is challenging for high dimension parameter spaces. The main advantage of URP is its simplicity, and that it does not require an optimisation algorithm.

The *two-dimensional Profile Likelihood* (PL2) method provides the most information about the parameter domain. In particular, it is the only method presented in this paper which can diagnose parameter inter-dependency and identifiability, as well as handle multimodal objective functions with local minima. Bootstrapping methods may show large dispersion in estimated parameters if parameters are non-identifiable, but the exhaustive exploration of the entire parameter space Θ offered by the PL2 method still provides more detailed and clear diagnostic conclusions. Since the method obtains highly descriptive profiles of combinations of parameters, this method provides the most detailed information about the parameter space Θ. Hence, if methods like PL1 or URP indicate problems with identifiability, it may be useful to apply the PL2 method to obtain a better analysis of the parameter space. Finally, the PL2 method is guaranteed to find the global optimum in Θ, within the accuracy allowed by the discretisation for the brute force search.

Repeatedly optimising the parameters with random initial guesses can be used to test the parameter optimisation procedure for sensitivity to the initial conditions. Additionally, this method is a useful tool for identification of local minima in the objective function. If there are multiple locally optimal solutions, this method will likely find them faster than the PL2 method, provided that the distribution of randomised initial conditions is dense enough, i.e., it needs a large enough number of repeated randomised initial conditions with subsequent optimisation of parameters such that at least one of the randomly drawn initial guesses will be close enough to the local optima to find them.

Bootstrapping Politis [2003] is perhaps the most intuitive way to obtain confidence regions, since it resembles the basic idea of computing coverage probabilities for multiple experiments Neyman [1937]. However, as the results have shown, if the dynamic information content in the data varies in time, block based bootstrapping can create pseudo data-sets that are *uninformative* w.r.t. parameter estimation and hence provide poor parameter identifiability. Subsequently, there can be *outlier* parameter estimates among the M iterations which affect the computation of mean parameters and the covariance. When there are variations in dynamic information content in the calibration data, special care should be taken when selecting the block lengths for bootstrapping. Regardless, bootstrapping is much faster than the PL2 method, and is therefore a useful alternative or augmentation to the PL2 method, in particular where computational resources and/or time is a challenge. Arguably, bootstrapping may also provide a more *realistic* estimation of the uncertainty of the parameters, provided the consistency of dynamic information in the calibration data is acceptable, since the method approximates running repeated experiments in a way that is similar to the idea of *coverage probability* calculation for confidence intervals. Due to its simplicity of implementation, bootstrapping methods may be preferable as an initial estimate of the uncertainty of estimated parameters.

Finally, a moving window combined with the PL1, or the Hessian method, can be used to test for consistency in dynamic information w.r.t. a particular model. Since this method, especially based on the PL1 method, is somewhat time consuming, it is most useful as a diagnostic tool to test for sources of diverging results in other methods, such as block based bootstrapping.

4 Conclusion

In this paper, a number of different methods for parameter estimation and analysis has been presented. Two test cases, a simple first order model with simulated

data, and a thermal network building grey-box model with measurement data from a physical building, was used to demonstrate the application of these methods.

The main results from these two test cases are, firstly, demonstrating the usefulness of one- and two-dimensional *Profile Likelihood* Raue et al. [2009]. These methods obtain *descriptive profiles* for each parameter, which can both estimate the *uncertainty* of the parameter estimate, diagnose the *identifiability* of the parameters and test for presence of local minima. The two-dimensional Profile Likelihood was shown to be particularly useful for detecting *over-parametrisation* for the second test case. Further, the one dimensional profile likelihood method was used with a moving window to check the *consistency* of dynamic information, and subsequently the identifiability and estimation uncertainty of the parameters as a function of *time*, with respect to a specific model structure. The latter was shown to be useful in combination with *block based bootstrapping*, to test for segments of data that are *uninformative* w.r.t. parameter estimation.

For the first test case, six different simulated data-sets were used. Of these six sets, the simple input step and the Pseudo Random Binary Sequence with $0.5s$ bit length gave the lowest overall estimation uncertainty. However, since the step data-set contains significant segments of data in which the system is in steady state, and hence produce non-identifiable parameters, the use of block based bootstrapping method results introduce outliers in the parameter estimates which significantly inflate the covariance of the mean parameter estimate. Hence, the interesting conclusion for this test case is that the data-set which produces the *lowest* estimation uncertainty for the Profile Likelihood and Hessian based method gives the *highest* uncertainty for the block based bootstrap method. Hence, what methods to use is also affected by the *dynamic information* content in the calibration data, and consequently the experimental design used to obtain that data, in addition to the application requirements and desired representation of resulting parameters.

References

Akaike, H. Information theory and an extension of the maximum likelihood principle. In *Selected papers of Hirotugu Akaike*, pages 199–213. Springer, 1998. doi:10.1007/978-1-4612-1694-0_15.

Bacher, P. and Madsen, H. Identifying suitable models for the heat dynamics of buildings. *Energy and Buildings*, 2011. 43(7):1511 – 1522. doi:10.1016/j.enbuild.2011.02.005.

Bentley, J. P. *Principles of measurement systems*. Pearson education, 2005.

Berthou, T., Stabat, P., Salvazet, R., and Marchio, D. Development and validation of a gray box model to predict thermal behavior of occupied office buildings. *Energy and Buildings*, 2014. 74:91–100. doi:10.1016/j.enbuild.2014.01.038.

Bohlin, T. and Graebe, S. F. Issues in nonlinear stochastic grey box identification. *International journal of adaptive control and signal processing*, 1995. 9(6):465–490. doi:10.1002/acs.4480090603.

Brastein, O., Perera, D., Pfeiffer, C., and Skeie, N.-O. Parameter estimation for grey-box models of building thermal behaviour. *Energy and Buildings*, 2018. 169:58 – 68. doi:10.1016/j.enbuild.2018.03.057.

Brastein, O. M., Lie, B., Sharma, R., and Skeie, N.-O. Parameter estimation for externally simulated thermal network models. *Energy and Buildings*, 2019a. 191:200–210. doi:10.1016/j.enbuild.2019.03.018.

Brastein, O. M., Sharma, R., and Skeie, N.-O. Sensor placement and parameter identifiability in grey-box models of building thermal behavior. In *Proceedings of The 60th Conference on Simulation and Modelling (SIMS 60), 13-16 August 2019, Västerås, Sweeden*. Linköping University Electronic Press, 2019b.

Deconinck, A.-H. and Roels, S. Is stochastic grey-box modelling suited for physical properties estimation of building components from on-site measurements? *Journal of Building Physics*, 2017. 40(5):444–471. doi:10.1177/1744259116688384.

Efron, B. Bootstrap Methods: Another Look at the Jackknife. *The Annals of Statistics*, 1979. 7(1):1–26. doi:10.1007/978-1-4612-4380-9_41.

Ergon, R. and Di Ruscio, D. Dynamic system calibration by system identification methods. In *European Control Conference (ECC), 1997*. IEEE, pages 1556–1561, 1997. doi:10.23919/ECC.1997.7082324.

Farrell, J. A. and Polycarpou, M. M. *Adaptive approximation based control: unifying neural, fuzzy and traditional adaptive approximation approaches*, volume 48. John Wiley & Sons, 2006.

Ferrero, C. S., Chai, Q., Dueñas Díez, M., Amrani, S. H., and Lie, B. Systematic analysis of parameter identifiability for improved fitting of a biological wastewater model to experimental data. *Modeling, Identification and Control*, 2006. 27(4):219. doi:10.4173/mic.2006.4.2.

Fux, S. F., Ashouri, A., Benz, M. J., and Guzzella, L. EKF based self-adaptive thermal model for a passive house. *Energy and Buildings*, 2014. 68:811–817. doi:10.1016/j.enbuild.2012.06.016.

Hoos, H. H. and Stützle, T. *Stochastic local search: Foundations and applications*. Elsevier, 2004.

Jazwinski, A. H. *Stochastic processes and filtering theory*. Dover Publications, Inc, 1970.

Johansson, R. *System Modeling and Identification*. Information and system sciences series. Prentice Hall, 1993.

Johnson, R. and Wichern, D. *Applied Multivariate Statistical Analysis*. Applied Multivariate Statistical Analysis. Pearson Prentice Hall, 2007.

Juhl, R., Møller, J. K., Jørgensen, J. B., and Madsen, H. Modeling and prediction using stochastic differential equations. In *Prediction Methods for Blood Glucose Concentration*, pages 183–209. Springer, 2016a. doi:10.1007/978-3-319-25913-0_10.

Juhl, R., Møller, J. K., and Madsen, H. ctsmr-Continuous Time Stochastic Modeling in R. *arXiv preprint arXiv:1606.00242*, 2016b.

Killian, M. and Kozek, M. Ten questions concerning model predictive control for energy efficient buildings. *Building and Environment*, 2016. 105:403–412. doi:10.1016/j.buildenv.2016.05.034.

Kristensen, N. R. and Madsen, H. Continuous time stochastic modelling. *Mathematics Guide*, 2003. pages 1–32.

Kristensen, N. R., Madsen, H., and Jørgensen, S. B. Parameter estimation in stochastic grey-box models. *Automatica*, 2004. 40(2):225–237. doi:10.1016/j.automatica.2003.10.001.

Kullback, S. A Note on Neyman's Theory of Statistical Estimation. *The Annals of Mathematical Statistics*, 1939. 10(4):388–390. URL https://www.jstor.org/stable/2235617.

Kunsch, H. R. The jackknife and the bootstrap for general stationary observations. *The Annals of Statistics*, 1989. pages 1217–1241. URL https://www.jstor.org/stable/2241719.

Lie, B. Model uncertainty and control consequences: a paper machine study. *Mathematical and Computer Modelling of Dynamical Systems*, 2009. 15(5):463–477. doi:10.1080/13873950903375452.

Ljung, L. *System Identification: Theory for the User*. Prentice Hall information and system sciences series. Prentice Hall PTR, 1999.

Lodhi, H. and Gilbert, D. Bootstrapping parameter estimation in dynamic systems. In *International Conference on Discovery Science*. Springer, pages 194–208, 2011. doi:10.1007/978-3-642-24477-3_17.

Madsen, H. *Time series analysis*. Chapman and Hall/CRC, 2007.

Madsen, H. and Holst, J. Estimation of continuous-time models for the heat dynamics of a building. *Energy and buildings*, 1995. 22(1):67–79. doi:10.1016/0378-7788(94)00904-X.

Maiwald, T. and Timmer, J. Dynamical modeling and multi-experiment fitting with PottersWheel. *Bioinformatics*, 2008. 24(18):2037–2043. doi:10.1093/bioinformatics/btn350.

Meeker, W. Q. and Escobar, L. A. Teaching about approximate confidence regions based on maximum likelihood estimation. *The American Statistician*, 1995. 49(1):48–53. doi:10.1080/00031305.1995.10476112.

Murphy, S. A. and Van der Vaart, A. W. On profile likelihood. *Journal of the American Statistical Association*, 2000. 95(450):449–465. doi:10.1080/01621459.2000.10474219.

Neyman, J. Outline of a theory of statistical estimation based on the classical theory of probability. *Philosophical Transactions of the Royal Society of London. Series A, Mathematical and Physical Sciences*, 1937. 236(767):333–380. doi:10.1098/rsta.1937.0005.

Nocedal, J. and Wright, S. *Numerical optimization*. Springer Science & Business Media, 2006.

Pohjanpalo, H. System identifiability based on the power series expansion of the solution. *Mathematical Biosciences*, 1978. 41(1):21–33. doi:10.1016/0025-5564(78)90063-9.

Politis, D. N. The impact of bootstrap methods on time series analysis. *Statistical Science*, 2003. pages 219–230. URL https://www.jstor.org/stable/3182852.

Politis, D. N. and Romano, J. P. The stationary bootstrap. *Journal of the American Statistical association*, 1994. 89(428):1303–1313. doi:10.1080/01621459.1994.10476870.

Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P. *Numerical recipes in C++*, volume 3. Cambridge University Press, 2007.

Raue, A., Kreutz, C., Maiwald, T., Bachmann, J., Schilling, M., Klingmüller, U., and Timmer, J. Structural and practical identifiability analysis of partially observed dynamical models by exploiting the profile likelihood. *Bioinformatics*, 2009. 25(15):1923–1929. doi:10.1093/bioinformatics/btp358.

Rosen, R., Wichert, G. v., Lo, G., and Bettenhausen, K. D. About the importance of autonomy and digital twins for the future of manufacturing. *IFAC-PapersOnLine*, 2015. 48(3):567 – 572. doi:10.1016/j.ifacol.2015.06.141. 15th IFAC Symposium onInformation Control Problems inManufacturing.

Rossi, R. J. *Mathematical Statistics: An Introduction to Likelihood Based Inference*. John Wiley & Sons, 2018.

Runge, C. Ueber die numerische Auflösung von Differentialgleichungen. *Mathematische Annalen*, 1895. 46(2):167–178. doi:10.1007/BF01446807.

Simon, D. *Optimal state estimation: Kalman, H infinity, and nonlinear approaches*. John Wiley & Sons, 2006.

Venzon, D. and Moolgavkar, S. A method for computing profile-likelihood-based confidence intervals. *Applied statistics*, 1988. pages 87–94. doi:10.2307/2347496.

Wang, L. *Model predictive control system design and implementation using MATLAB®*. Springer Science & Business Media, 2009.

Wilks, S. S. The Large-Sample Distribution of the Likelihood Ratio for Testing Composite Hypotheses. *The Annals of Mathematical Statistics*, 1938. 9(1):60–62. URL https://www.jstor.org/stable/2957648.