ROBUST PARAMETER ESTIMATION OF CHAOTIC SYSTEMS

SEBASTIAN SPRINGER*, HEIKKI HAARIO AND VLADIMIR SHEMYAKIN
Lappeenranta University of Technology
Department of Computational Engineering
Lappeenranta, 53850, Finland

Finnish Meteorological Institute
Atmospheric Remote Sensing Group
Helsinki, 00560, Finland

LEONID KALACHEV AND DENIS SHCHEPAKIN
University of Montana, Department of Mathematical Sciences, 59812, Missoula
Department of Mathematical Sciences
Missoula, 59812, Montana

(Communicated by Matti Lassas)

ABSTRACT. Reliable estimation of parameters of chaotic dynamical systems is a long standing problem important in numerous applications. We present a robust method for parameter estimation and uncertainty quantification that requires neither the knowledge of initial values for the system nor good guesses for the unknown model parameters. The method uses a new distance concept recently introduced to characterize the variability of chaotic dynamical systems. We apply it to cases where more traditional methods, such as those based on state space filtering, are no more applicable. Indeed, the approach combines concepts from chaos theory, optimization and statistics in a way that enables solving problems considered as ‘intractable and unsolved’ in prior literature. We illustrate the results with a large number of chaotic test cases, and extend the method in ways that increase the accuracy of the estimation results.

1. Introduction. A number of the recently published papers discuss the difficulties of parameter estimation which are introduced by the chaoticity of the models (see, e.g., [35], [37], [26], [29] for examples). The situation is usually dealt with as a ‘black box’ least-squares fitting problem. However, this approach is not appropriate for such systems on large time interval integrations, since a least squares solution does not even exist in the same sense as it exists for deterministic models. A new integration of the same system with the very same parameter values and initial values, but using a different solver or slightly different solver tolerances, typically leads to a totally different trajectory after an initial time interval over which the trajectory is predictable.

The problem of chaoticity is trivially avoided if, in addition to well-known initial values, enough data is available on a short enough time interval where the system remains predictable. However, this is hardly the situation with higher dimensional systems, such as turbulent flows or weather models. But the idea can be employed to

2010 Mathematics Subject Classification. 62F15, 62F86, 68U01, 37D45, 34C28, 65C05.

Key words and phrases. Chaotic dynamical systems, parameter estimation, stochastic optimization, Bayesian inference, Markov Chain Monte Carlo (MCMC).

The first author is supported by the CoE, Academy of Finland, decision number 312 122.

* Corresponding author: Sebastian Springer.

©2019 American Institute of Mathematical Sciences
construct likelihoods based on Kalman filtering. In particular, one divides the total integration time interval into subintervals, often called the assimilation windows, which are short enough for performing deterministic (predictable) simulations. The construction of a trajectory using the measurement data is then followed by the adjustment of a state vector at the start of each assimilation window. This process ‘integrates out’ the state vector, and can be used to construct an expression that only depends on data and parameters, i.e., the filter likelihood. While standard for linear time series and stochastic differential equations \cite{30}, \cite{7}, this approach leads to some tuning issues for chaotic systems; see \cite{13} and \cite{14}. We note that the use of predictable assimilation intervals is the essence of weather prediction algorithms, albeit basic Kalman filtering methods are not applicable due to memory issues in high dimensions, and so 4D-VAR methods are employed; see \cite{17}.

Another obvious approach to improve the estimation of the chaotic system parameters is to use averaging. In \cite{15} we constructed a cost function based on zonal and temporal averages, simulated by the ECHAM5 climate model. While the stochasticity of the cost function was damped enough to get usual Markov chain Monte Carlo (MCMC) sampling technically working, this likelihood did not constrain the model parameters: naive summary statistics do not characterize the specific geometry of an underlying chaotic attractor in a meaningful way.

We show here that it is possible to construct a likelihood free of the above mentioned pitfalls, supposing that long enough time series data are available. In \cite{12} a distance concept was introduced for chaotic trajectories with the goal of parameter sensitivity analysis. The purpose was to characterize the ‘internal’ variability of a given chaotic system with known system parameter values, in order to distinguish trajectories that deviate form this variability. The focus of the current paper is on providing a robust solution to the chaotic system estimation problem by using this distance concept. In particular, we provide: (i) the estimation of the chaotic system parameters from noisy data, including cases where the initial state values are unknown and the initial model parameters lie far away from those producing the attracting chaotic manifold (we note that in \cite{12} the parameter values of a chaotic system were assumed to be known as the task was only to estimate the corresponding variability regions for these parameters); (ii) extensions of the distance concept, resulting in several improvements: more accurate confidence intervals for the estimated parameter values, estimating the dynamics, including the cases where the system involves vastly different time scales, or where the parameter estimation needs to be performed simultaneously for coupled chaotic and deterministic parts of a system. With several examples of chaotic systems we demonstrate how this likelihood construction allows for a robust parameter estimation, together with subsequent MCMC sampling of the parameter posteriors. We note that the proposed approach works even if the time intervals between observations are too long for the filtering based approaches to be applicable. Indeed, while the use of summary statistics to create cost functions or likelihoods is not new as such, see, e.g., \cite{1}, \cite{22}, \cite{36}, \cite{15}, the present approach combines concepts from chaos theory, optimization and statistics in a novel way that allows us to solve problems considered as ‘intractable and unsolved’ in prior literature; see, e.g., \cite{29}.

The rest of the paper is organized as follows. In Methods we discuss all the steps needed for performing parameters’ uncertainty quantification of chaotic dynamical systems. In particular, we discuss how the correlation integral concept can be modified to create an empirical likelihood from a given time series data. In
Extended feature vectors we extend the approach in a way that allows one to characterize the dynamics of a chaotic system. For the systems exhibiting ‘slow’ chaotic dynamics and ‘fast’ transitions to a chaotic manifold we show how to determine if a point belonging to a trajectory lies in a vicinity of a chaotic attractor or away from it. These results are important for identification of the portions of the trajectories on which the data must be collected to estimate the coefficient responsible for the ‘fast’ dynamics. Also, the algorithms discussed below require the randomization of the initial conditions to produce a set of trajectories belonging to an attractor; we indicate how to check if the initial conditions are located on or in a close vicinity of an attractor. The Results of the numerical experiments starts with the Lorenz63 system. We demonstrate the application of the approach for estimating the parameters from sparse data, with poor initial guesses for parameter values and unknown initial values of the state vector. In Lorenz63 with dynamics included, we present the results and discuss the guidelines for implementation. In Further numerical experiments we apply the method to the analysis of numerous well-known chaotic systems. We show how the proposed extension greatly improves the parameter identification, especially, in more complex cases. Brief discussion summarizing the results and addressing the next steps of the analysis and possible applications are included in Discussion and conclusions. The Appendix contains further details of the optimization method used, the derivation of some formulas discussed in the different time scales part of the text and the set up of all the numerical experiments.

2. Methods.

2.1. Parameter estimation. The procedure of parameter estimation in a standard, non-chaotic setting consists of three steps: constructing first the cost function as a statistical likelihood, based on known or assumed noise statistics of the measurement. Next, finding the maximum likelihood or MAP (maximum a posterior point) parameter estimate, by using some numerical optimization routine. Finally, finding the full posterior parameter distribution of the unknown, typically by the use of sampling methods as provided with several MCMC (Markov chain Monte Carlo) methods.

The main difficulty of estimating parameter values of a chaotic dynamical system is related to the fact that, indeed, no unique numerical solutions exist: even if the model parameter values and initial state values are fixed, the trajectories produced by numerical integrations with different numerical solvers, or slightly different solver tolerance choices, differ significantly on sufficiently long integration intervals. So, the standard construction of a likelihood function is not available. We will present here an alternative that enables parameter estimation in full analogy with the standard approach. However, the special nature of the likelihood construction imposes some requirements on the algorithms to be used.

We reiterate the basic ideas of the approach used in [12] and introduce the modifications required for the task of parameter estimation.

2.2. Correlation integral likelihood. Denote by \( s = s(t, \theta, x) \) a state vector \( s \) of a dynamical system, that depends on time \( t \), parameters \( \theta \) and other inputs \( x \) such as, e.g., the initial values. The measurements of the system \( s \) at the time points \( t_1, \ldots, t_N \) are denoted by \( s_i = s(t_i, \theta, x) \), \( i = 1, \ldots, N \). We consider two different trajectories \( s = s(t, \theta, x) \) and \( \bar{s} = s(t, \bar{\theta}, \bar{x}) \), obtained for different parameter values
and inputs sets, \((\theta, x)\) and \((\tilde{\theta}, \tilde{x})\), and evaluated at \(N\) time points \(t = \{t_1, \ldots, t_N\}\).

For \(R > 0\), the modified correlation sum is defined as

\[
C(R, N, \theta, x, \tilde{\theta}, \tilde{x}) = \frac{1}{N^2} \sum_{i,j \leq N} \#(\|s_i - \tilde{s}_j\| < R),
\]

where \(\#(\|s_i - \tilde{s}_j\| < R)\) denotes the number of points of the different trajectories, lying within a sphere of radius \(R\) from each other. In the case \(\tilde{\theta} = \theta\) and \(\tilde{x} = x\) the formula reduces to the well known definition of correlation sum. However, we are not interested in the small-scale limit \(R \to 0\) as in the classical definition of the correlation dimension, but use the expression \((1)\) to characterize the distance between different trajectories at all relevant scales \(R\). More specifically, we can define a likelihood for a given dynamical system with a fixed parameter value \(\theta\) as follows. We fix a set of radii \(R_k, k = 1, \ldots, M\) and consider a vector \(y\) of dimension \(M\) with components from \((1)\) with \(\tilde{\theta} = \theta\).

\[
y_k = C(R_k, N, \theta, x, \tilde{x}).
\]

For randomized values \(\tilde{x} \neq x\) the vector \(y\) is stochastic, with components that consists of averages. So by the Central Limit Theorem \(y\) can be expected to be Gaussian. Indeed, \((2)\) gives the empirical cumulative distribution function (Ecdf) for the respective set of distances, with the values \(R_k\) used as bins. The basic form of the Donsker’s theorem states that the empirical distribution functions asymptotically tend to a Brownian bridge. In a more general setting, close to one employed here, the normality was established by Borovkova et.al. [2]. So the expression \((2)\) provides a summary statistics that maps samples from a chaotic attractor into a Gaussian feature vector.

In practice, one can compute a set of Ecdf vectors of the distances \((2)\) by a training set obtained with a sufficient number of pairs \(\tilde{x} \neq x\). By calculating the mean \((\bar{y})\) and covariance \((\Sigma)\) of the \(y\) vectors, the Correlation integral likelihood (CIL) is obtained as the expression \(\exp\left(-\frac{1}{2}(y - \bar{y})^T \Sigma^{-1} (y - \bar{y})\right)\). Note that a similar synthetic likelihood construction was used in [36] for different criteria.

Next, let us discuss the construction of a similar likelihood function for the task of estimating unknown parameter values by data. Instead of creating the training set by simulations with perturbed values of \(x\), we resort to subsampling of the measurements. Assume that we have a time series of \(N_o\) measurements, large enough to cover all the characteristic features of the underlying chaotic dynamics. We create a training set for the CIL by sampling the representative subsets of length \(N\) of these data. The approach requires that all trajectories used in \((2)\) contain samples that cover the underlying attractor in a sufficient way. For this we must assume that \(N_o\) is large enough so that the subsamples of lengths \(N\) with this property can be constructed. Systems with rarely occurring events might yield sample sets that do not capture all such events. While this is not a serious issue in any of the numerous cases considered in the present work it may arise in more complicated situations. This question will be discussed in more detail in future elsewhere.

The bin values are given by \(R_i = R_0 b^{-i}, i = 1, \ldots, M\). The ‘tuning’ parameters of our method, \(R_0, b\) and \(M\), can be selected in a straightforward manner. \(R_0\) is set to slightly exceed the maximum value of the distances computed by the recipe described above. The smallest radius \(R_M = R_0 b^{-M}\) should be large enough so that for all the feature vectors in the training set the spheres with the smallest radius

Inverse Problems and Imaging Volume 13, No. 6 (2019), 1189–1212
contain neighbouring points. The remaining ‘free’ parameter is the number of bins \( M \) used to create the empirical cumulative functions. But the choice of \( M \) is the usual trade-off always present in constructing histograms or cumulative distribution functions. Once \( M \) and \( R_M \) are selected, the base value \( b \) can be found by solving \( R_M = R_0 b^{-M} \).

The normality of the ensuing distribution can be numerically checked using the \( \chi^2 \) distribution test; see the Results section below. Note that the averaging process used in the construction of the likelihood diminishes the variability of the likelihood function to the extent that the optimization and MCMC sampling typically perform without major problems. However, even if the chaotic variability is ‘tamed’, the evaluation of the cost function is still stochastic, and appropriate algorithms are needed for performing the maximum likelihood optimization.

2.3. Optimization: Population methods. As the Gaussian likelihood is constructed, we can search for its maximum likelihood point, and perform the subsequent MCMC runs for estimating the parameter posterior distributions.

This implies that the optimization method used to find the maximum likelihood point should be robust with respect to non-smoothness. Moreover, as we may start the parameter estimation with the initial guesses taken quite far away from the optimal values, and perform each model integration with randomized initial state values, the optimizer should have robust convergence properties. For these reasons we use here a variant of the Differential Evolution (DE) algorithm, a population-based optimization method. Differential Evolution belongs to the class of Evolutionary Algorithms (EAs), proposed in [33]; also see [27]. It consists of four main steps, initialization, mutation, crossover and selection, which are applied sequentially to an ensemble or ‘population’ of parameter values until certain criteria are met. The name of DE comes from the particular way in which a ‘mutation’ is performed, in particular, the new candidates are obtained by means of scaled differences of vectors added to other vectors.

DE was initially designed for deterministic problems, but slight modifications to the original formulation enable us to use it as a robust optimizer for stochastic problems as well [31]. An issue in optimizing a stochastic cost function is the risk of finding a false optimal value simply due to the noise, i.e., the larger the noise - the more likely it is to ‘freeze’ at a non-optimal parameter value. The risk is diminished by the use of a population of parameters. Nevertheless, a common pitfall of population methods is the ‘shrinkage’ of the parameter values to a few or one point. Here we avoid this by a recalculation step: the cost function values are updated for all the parameter values in the current population. To save CPU, however, this can be done only at some selected iteration steps. Also, we note that with a stochastic cost function we do not even aim at convergence to a single maximum likelihood point, but instead want the values to fluctuate around the ‘optimal’ value. In the evolutionary language, this means preserving the diversity among the population. Indeed, the recalculation step ensures the diversity. While the optimization result is heuristic, the final parameter population values turn out to provide a very effective initial proposal distribution for a rigorous MCMC sampling, see the discussion below.

Another crucial reason to use DE here is the ease of parallelization, as each population member can be run independently from others, and synchronization is only needed at the selection step. This is beneficial, especially, in the case of heavy CPU models.
The DE algorithm can be implemented in various ways. Except for the recalculation step we follow one of the standard variants. All the details are given in the Appendix 1.

2.4. **Uncertainty quantification: MCMC.** After the convergence of the DE optimization we can study the parameter distributions by MCMC. Note that the sampling of a stochastic likelihood function can be interpreted as a pseudomarginal sampling, see [12].

In the examples we will see that, as an additional benefit of the diversity preservation, we may use the final part of the history of the optimization to create an effective initial proposal distribution for the adaptive MCMC methods (the AM or DRAM algorithms, see [10], [11]) used in the posterior sampling step. The MCMC process starts with the mean value of the final generation of DE optimization and the proposal distribution selected as the Gaussian distribution with the covariance calculated from several previous generations. Such a choice of the proposal distribution provides an effective acceptance rate from the beginning, and supports subsequent adaptation of the proposal distribution. Indeed, the role of the heuristic DE algorithm is to find good initial values for the rigorous posterior estimation by MCMC methods.

One may anticipate that the final DE populations could provide a good initial proposal for MCMC. As shown in the Results section, even more can be true: the DE samples give a good approximation of the parameter posterior distribution itself. Naturally, this is case dependent and should not be expected in all situations.

2.5. **Extended feature vectors: Dynamics and different time scales.** The key idea in the above form of the estimation approach is to consider the computed or measured state vector values as a point cloud sample from the underlying fixed attractor, and to construct a likelihood based on the CIL feature vector. While the mapping from the point cloud to the CIL feature vector has preferable properties such as the Gaussianity, it also inevitably loses information. Indeed, the order of the points is irrelevant, as only the Ecdf of all the pairwise distances is used. Moreover, as only the phase space values are used, the dynamics of the system is lost. Here we introduce ways that enable the estimation of the dynamics of the systems. Note that in many applications dynamical systems may also contain vastly different time scales. So, we discuss both the ‘overall’ time evolution of a system, and ways to deal with systems consisting of fast or slow subsystems.

In order to enhance the information extracted from the measured point cloud, we can introduce several constructions in addition to the phase space values such as using the concept of embedded dimension. Here we focus on how to include the time behaviour of the system. We extend the state vector by the time derivative, i.e., consider the extended state vector \( \tilde{s} = (s, \dot{s}) \). During a numerical integration the values of \( \dot{s} \) are easily obtained by the right hand side of the differential equation system. The respective experimental values by noisy measurements must be calculated judiciously, as the task of numerical differentiation of noisy data is a well-known ill-posed problem. In our test cases we use just a basic statistical rule for selecting the time difference \( \Delta t \) between the noisy observation points: the difference of observed consecutive state values should be 2 – 3 times larger than the noise level of the data. Our examples demonstrate that the accuracy of parameter estimation is clearly improved, indeed, dramatically in many cases, by introducing such derivative information. Note that we do not need overall dense measurement
points; sparse sets of pairs \((t_i, t_i + \Delta t_i)\) are enough, and they are used in the example cases below.

The state values and the respective derivatives may have quite different scales, so we create different feature vectors for each of them. The final stochastic feature vector is obtained by concatenating the two vectors. The Gaussianity of the combined vector can again be confirmed in the same way as earlier. We give several examples detailing these steps below in the Results section.

Next, let us discuss of the reliable parameter estimation for chaotic systems with dynamics taken into account for the more complex situations, where different parts of the system may have vastly different time scales. In particular, assume that the fast dynamics corresponds to a stable transition of the solution trajectories starting “outside” of the “slow” chaotic manifold and reaching a quasi-steady state with the slow manifold.

For the original system of equations containing both fast and slow variables we may want to estimate the coefficient for both, the slow and the fast, parts of the dynamical system as a part of one procedure combining the classical deterministic and the chaotic system parameter identification approaches. However, the parameters of the fast transition cannot be identified as part of the CIL likelihood, but should be estimated from separate measurements taken before the solution trajectory reaches some vicinity of a slow manifold and then continues staying in a quasi-steady state with the manifold. To do that we need to make sure that the point with which we start the ‘fast’ parameter fit (as well as the data which we may want to use in the parameter identification process) indeed lies away from the chaotic attractor. On the other hand, in the CIL approach we should check that after randomization the initial values of the state variables, the points taken from the trajectory are, indeed, located on the slow chaotic manifold.

Both situations call for the methods that enable one to estimate if a given point is located close to or away from a ‘slow’ attractor. Below, we discuss the procedure, initially introduced in [24], that allows one to check if a particular point belonging to a solution trajectory, has a slow manifold (chaotic or deterministic) in its vicinity.

Although the procedure works in a more general form, where the original system contains several fast variables in addition to the slow ones here, for the sake of clarity, we discuss a special case of the system with a single fast variable which adjusts to the behavior of one chaotic variable, or a combination of chaotic variables; in particular, we may note that every system presented in Appendix 3 can be written in the following form:

\[
\dot{x} = f(x),
\]

where the vector functions on the right-hand side of (3) are just the right-hand sides of the equations shown in Appendix 3; in (3),

\[
x(t) = (X(t), Y(t), Z(t))^* \\
= (x_1, x_2, x_3)^*
\]

and

\[
f(x) = (F(X, Y, Z), G(X, Y, Z), H(X, Y, Z))^* \\
= (f_1, f_2, f_3)^*,
\]

here “*” stands for transpose. Now we will only consider the chaotic systems with the differentiable right-hand sides; since the right-hand sides for the majority of
the models presented in Appendix 3 are polynomials, this condition is obviously satisfied for them.

Consider an extended system:

\[
\begin{aligned}
\dot{x} &= f(x), \\
\dot{W} &= -K \cdot (W - X),
\end{aligned}
\]

where \( W \) is a “fast” scalar variable and \( K > 0 \) is a large constant associated with characteristic time of adjustment of \( W(t) \) to \( X(t) \). Before we may apply the results of [24] we need to re-write (4) in a special form considered in [24]. First, let us change the variable in (4):

\[
V(t) = W(t) - X(t),
\]

and thus,

\[
\dot{V} = \dot{W} - \dot{X} = -K \cdot V - F(X, Y, Z).
\]

Now we can introduce the notation for a new vector-function

\[
y(t) = (X(t), Y(t), Z(t), V(t))^* = (x(t), V(t))^* = (y_1, y_2, y_3, y_4)^*.
\]

We can write the system for \( y(t) \) as \( \dot{y} = g(y) \), where

\[
g(y) = (F(X, Y, Z), G(X, Y, Z), H(X, Y, Z), -K \cdot V - F(X, Y, Z))^*.
\]

Let us fix some point \((y(t_0), t_0)\) belonging to the solution trajectory, and check if there exists an attractive locally invariant manifold near this point. If the conditions guaranteeing the existence of such manifold are satisfied, then the chosen point belongs to the trajectory lying on the manifold and, thus, the system may be reduced to a lower dimensional one (eliminating an explicit dependence of the solution on \( K \) on sufficiently long the time intervals). If, on the other hand, these conditions are not satisfied, then the chosen point is located outside the slow manifold and the reduction to a lower dimensional system in this case is not possible; the full system formulation needs to be considered to describe the dynamics of the solution in the vicinity of such point (which would mean that the value of \( K \) can be estimated from the data collected on the time intervals of the order \( O(1/K) \)).

In what follows we will use the upper index \(^{\text{circ}}\) to denote the values of various expressions evaluated at \((y(t_0), t_0)\). We re-write the system \( \dot{y} = g(y) \) in an equivalent form:

\[
\dot{y} = g^0 + J^0 \cdot (y - y_0) + \tilde{g}(y, y_0);
\]

here \( J^0 \) is the Jacobian matrix of (6), i.e., \( J = g_{yy} \), evaluated at \((y(t_0), t_0)\), and

\[
\tilde{g}(y, y_0) = g(y) - g^0 - J^0 \cdot (y - y_0) = O(|y - y_0|^2).
\]

The notation \( | \cdot | \) is used to denote the Euclidean norm for vectors; in case of matrices, this will denote the matrix norm induced by the Euclidean vector norm: i.e., for a matrix \( A \) we have that \(|A| = \sqrt{\rho(A^* \cdot A)}\), where \( \rho \) is the spectral radius of \( A \). The \((4 \times 4)\) Jacobian matrix \( J^0 \) will have four eigenvalues of which...
three are the eigenvalues of the \((3 \times 3)\) matrix \(f_x^0\); for illustrative purposes, and without loss of generality, we assume that they are all real distinct: \(\lambda_1 \neq \lambda_2 \neq \lambda_3\); similar procedure works in a more general case as well. The fourth eigenvalue is \(\lambda_4 = -K\), with \(K \gg 1\).

Let us introduce the variables transformation \(\mathbf{y} = \mathbf{y}_0 + \mathbf{T} \cdot \mathbf{u}\), where \(\mathbf{T}\) is the matrix transforming \(\mathbf{J}_0^0\) into a diagonal form. Applying this change of variables to system (7), we obtain (see the details in Appendix 2; notations \(\Lambda_0^0\) and \(\Lambda_0^1\) used below denote the matrices of eigenvalues; they are defined in the formula eq. (S1) of Appendix 2)

\[
\dot{\mathbf{u}} = \mathbf{T}^{-1} \mathbf{g}_0^0 + \Lambda_0^0 \cdot \mathbf{u} + \mathbf{T}^{-1} \tilde{\mathbf{g}}(\mathbf{y}_0 + \mathbf{T} \mathbf{u}, \mathbf{y}_0),
\]

which, when we use the notations

\[
\mathbf{u} = (u_1, u_2, u_3, u_4)^* = (\hat{\mathbf{u}}, u_4)^*,
\]

\[
\mathbf{T}^{-1} \mathbf{g}_0^0 = (\hat{g}_1^0, \hat{g}_2^0, \hat{g}_3^0, \hat{g}_4^0)^* = (\hat{g}_0^0, \hat{g}_0^4)^*,
\]

and

\[
\mathbf{T}^{-1} \tilde{\mathbf{g}}(\mathbf{y}_0 + \mathbf{T} \mathbf{u}, \mathbf{y}_0) = (\hat{h}_1, \hat{h}_2, \hat{h}_3, \hat{h}_4)^* = (\hat{h}(\hat{\mathbf{u}}, \mathbf{y}_0), \hat{h}_4(\hat{\mathbf{u}}, \mathbf{y}_0))^*,
\]

is equivalent to

\[
\begin{aligned}
\dot{\hat{u}}' &= \Lambda_1^0 \cdot \hat{\mathbf{u}} + \hat{\mathbf{g}}^0 + \hat{\mathbf{h}}(\hat{\mathbf{u}}, \mathbf{y}_0), \\
\dot{u}_4 &= -K \cdot u_4 + \frac{1}{K} \hat{g}_4^0 + \frac{1}{K} \hat{h}_4(\hat{\mathbf{u}}, \mathbf{y}_0),
\end{aligned}
\]

(8)

We note that from a special structure of the system (4) it follows that the vector-function \((\hat{h}, \hat{h}_4)^*\) does not depend on \(u_4\)! After re-writing the second equation of (9) as

\[
\frac{1}{K} \dot{u}_4 = -u_4 + \frac{1}{K} \hat{g}_4^0 + \frac{1}{K} \hat{h}_4(\hat{\mathbf{u}}, \mathbf{y}_0),
\]

we can easily check that (9) and (8) are exactly in the form for which the results in [24] were derived. According to the simplified algorithm presented in [24] to verify if there exists a “slow” attracting locally invariant manifold in the vicinity of the point \((\mathbf{y}(t_0), t_0)\) defined as a ball of radius \(\tilde{R}\) in the 4-dimensional phase space, we need to check the following inequality:

\[
\frac{1}{K} |\hat{g}_4^0| < \tilde{R}.
\]

(10)

If inequality (10) is satisfied, then the point \((\mathbf{y}(t_0), t_0)\) lies close to or on the slow manifold and the phase space of the system may be reduced from 4-dimensional to 3-dimensional (for \(K \gg 1\), which is equivalent to \(1/K \ll 1\), the slow manifold in the leading order approximation will correspond to \(u_4 = 0 + O(1/K)\)); if, on the other hand, (10) is not satisfied, then the point \((\mathbf{y}(t_0), t_0)\) lies outside \(\tilde{R}\)-vicinity of the manifold, and the reduction is not possible (which allows one to identify the value of parameter \(K\) if the data is collected at the appropriately (densely) chosen time points). An example illustrating this procedure is included in the Results section.
3. Results.

3.1. Example: Lorenz 63. We first demonstrate the steps described above and discuss the results for the classical example of the Lorenz 63:

\[
\dot{X} = \beta (Y - X); \quad \dot{Y} = X (\gamma - Z) - Y; \quad \dot{Z} = X \cdot Y - \alpha \cdot Z.
\]

As for the data signal, we select the values of \(X\) and \(Y\) at 8000 time points uniformly distributed over the time interval \([0, 80000]\), perturbed with Gaussian noise of the size corresponding to 5% of the signal. By purpose, we let the gap between the measurements be larger than the predictable time interval of the system. Figure 1 exhibits the initial stage of the analysis: an ensemble of 64 simulations with slightly varying initial values is shown. While the predictable time interval is roughly of length 7, the measurements are taken at intervals of length 10 apart. Examples of possible measurements are denoted in the picture (naturally, for the data signal the values of only one trajectory are selected). So, the measurement values are clearly impossible to predict over the assimilation window, which rules out the use of any of the standard filtering methods discussed in the Introduction. The likelihood function is constructed from the measured signal by a subsampling approach. We randomly select two disjoint sets of 2000 measurement vectors \(X_i, Y_i\) from the signal, compute the \(2000 \times 2000\) matrix of Euclidean distances between the vectors from the different sets, and create the empirical cumulative distribution function of the distances in the log-log scale to get one of our feature vectors. This is repeated around 2000 times.

In the present case, we select the number of radii to be \(M = 10\) and get \(R_0 = 2.51, b = 2.04\). Finally, we can check the Normality of the ensuing distribution. Figure 2b shows that the histogram preserves the \(\chi^2\) distribution, and Figure 2a exhibits the variability of the feature vectors of the training set. Now that the cost function is constructed, the DE optimizer is applied. Figure 2c shows the convergence to a
narrow region around the true parameter set values $\theta_0$ from a very broad range of initial guesses.

After the convergence we can study the parameter distributions by MCMC. The mean value of the final generations of the optimization and the respective covariance of the population members are used as the initial chain point and the initial proposal distribution for the MCMC sampling. In Figure 2d the 2D marginal posterior distributions of the chain obtained by AM and the last few generations of the DE population used to create the initial proposal distribution for the AM sampling method are presented.

3.2. **Lorenz 63 with dynamics included.** Let us assume that all the settings for the data and the computational setup of the previous example hold. In addition, we consider also the Euclidean distances between the velocity vectors $(X, Y)$. In the training set this requires that the additional measurements are available: the observation time points $t_i = 10 \cdot i, i = 1, ..., 8000$ are replaced with pairs $(t_i, t_i + \Delta t)$. The noise level is kept at 5%. The step $\Delta t$ must be large enough to allow an
estimate of the derivative by the noisy data. Other than that, the approach is not too sensitive with respect to the choice of a time step. In the present example a simple Runge-Kutta ODE solver is employed, with time step of 0.01. We choose $\Delta t = 0.1$, but note that any value in the range $0.1 < \Delta t < 0.8$ produces quite similar results.

![Log/log curve of the distances](image1)

![Log/log curve of the velocities](image2)

(A) Feature vectors in log-log scale. Distances between states and velocity vectors, respectively.

![Chi-squared test of the log/log Ecdf’s](image3)

(b) $\chi^2$ test of the log/log Ecdf’s in Figure 3a.

![Posterior distribution and DE proposal](image4)

(c) Lorenz 63: Posterior distribution and DE proposal. With dynamics included and also "K"-parameter estimated.

![Comparison posterior distribution](image5)

(d) Lorenz 63: Comparison posterior distribution with and without dynamics included.

**Figure 3.** Lorenz 63: Analysis performed after the addition of the second feature vector $(S, \dot{S})$.

We map the sets of derivative distances into a log/log curve, measured at $M = 10$ different decreasing radii with base $b_{vel} = 2.0$, see Figure 3a for plots of the feature vectors for both the state and the derivative values. To create a likelihood we concatenate the respective vectors into a $2M$ dimensional feature vector. It is then straightforward to check again the Gaussianity of the resulting stochastic vector, see Figure 3b for the numerical verification by the $\chi^2$ distribution test.

The parameter estimation proceeds exactly as before, only with the likelihood function now using the extended feature vector. Figure 3c illustrates a noticeable
improvement in the identification of the parameters after the derivative update to
the likelihood function. Other examples (see below) can show even much more
significant improvements.

The addition of the derivative values allows us to also estimate the overall speed of
time evolution of the system, something not available for the basic CIL version. We
demonstrate this by multiplying the right hand side of (11) by a constant $K = 10$
and estimating the four parameters $\alpha, \beta, \gamma, K$. In this case, especially, the final
DE population values are strikingly close to the MCMC posterior values. Note,
however, that the DE posterior approximation typically has a larger uncertainty, as
it consists of a relatively small sample of parameters, with no guarantee of ergodicity.
Finally, let us consider the Lorenz63 system (11) as an example to illustrate the
methodology of different time scales. The statement corresponds to the systems in
the general form (3).

The extended system, corresponding to (4), is:

$$
\begin{align*}
\dot{X} &= \beta(Y-X); & \dot{Y} &= X(\gamma-Z)-Y; & \dot{Z} &= X\cdot Y-\alpha\cdot Z; & \dot{W} &= -K(W-X),
\end{align*}
$$

where $K > 0$ is a large constant. Changing the variable (compare with (5)) $V(t) = W(t) - X(t)$, and rewriting the system (12) in the form (7), where $(X_0, Y_0, Z_0, V_0)$
is some chosen point of interest lying on the solution trajectory and $\tilde{X} = X - X_0$, $
\tilde{Y} = Y - Y_0$, $\tilde{Z} = Z - Z_0$, and $\tilde{V} = V - V_0$, gives

$$
\begin{pmatrix}
\dot{X} \\
\dot{Y} \\
\dot{Z} \\
\dot{V}
\end{pmatrix} =
\begin{pmatrix}
\beta(Y_0 - X_0) \\
X_0(\gamma - Z_0) - Y_0 \\
X_0Y_0 - \alpha Z_0 \\
-KV_0 - \beta(Y_0 - X_0)
\end{pmatrix} +
\begin{pmatrix}
-\beta & \beta & 0 & 0 \\
(\gamma - Z_0) & -1 & -X_0 & 0 \\
Y_0 & X_0 & -\alpha & 0 \\
\beta & -\beta & 0 & -K
\end{pmatrix}
\begin{pmatrix}
\tilde{X} \\
\tilde{Y} \\
\tilde{Z} \\
\tilde{V}
\end{pmatrix}
$$

Now, let us consider the choice of parameters: $\alpha = 8/3$, $\beta = 10$, $\gamma = 28$ and the
case were the variable $V$ is fast ($K = 1000$). We take such initial condition that the
trajectory starts on the attractor for the variables $X, Y, Z$ and off the attractor for
$V$ variable. Then we take a few points on the trajectory corresponding to different
values of $t$ and check the condition (10) for them. The results are presented in the
Figure 4. With respect to the chosen value of $R = 2$ the points corresponding to
t = 1.0025 and $t = 1.0035$, for which the condition (10) is satisfied, are considered
belonging to the $R$-vicinity of the chaotic attractor.

3.3. Further numerical experiments. To test the robustness of the approach
further, simulations similar to those presented above for the Lorenz system have
been performed for a large number of other chaotic systems. The respective equations,
together with the CIL parameters used to create the likelihood functions, are
listed in Appendix 3.

In all cases our approach leads to consistent results. We used the choice $M = 10$
for the number of bins for all cases, and obtained the values for $R_0, R_M$ and $b$ with
the same recipe as in the case of Lorenz 63 explained above. So we emphasize that
the choice of these constants does not typically require any elaborate case-dependent
hand tuning. However, one has to pay attention to selecting a long enough time
interval for the data collection, to ensure that all the ‘corners’ of the attractor
Figure 4. The solid line represents the projection of the solution trajectory onto an \((X,V)\) plane and the dashed line is the corresponding attractor \((V = 0)\). The solution is already on the attractor for \(Y\) and \(Z\) variables (the initial values were taken to be \(X(0) = 2.51, Y(0) = 2.51, Z(0) = 19.92, V(0) = -20\)). The selected points on the trajectory projection corresponding to different instants of time are marked with the stars \((t_1 = 1.0005, t_2 = 1.0015, t_3 = 1.0025,\) and \(t_4 = 1.0035\)). To make the results more visual, different scales were used for the \(X\) and \(V\) axes (otherwise the trajectory would have looked like a vertical line drawn to the \(V\) axis representing the attractor); as a result, for the same reason of making the illustration more comprehensible, we have drawn the ellipses with main axes \(1.3 \times 10^{-4}\) and 2 instead of circles of radius \(\tilde{R} = 2\). The corresponding values of \(1K_{\hat{g}_0}\) for the chosen time points \(t_1, t_2, t_3,\) and \(t_4\) are 13.67, 5.03, 1.85, and 0.68, respectively.

are represented in the training of the likelihood. Note that a natural requirement for successful simulations is that the system remains inside the same domain of attraction, both with respect to the initial values of the state vector and the model parameter values.

The number of parameters estimated for the cases listed in Appendix 3 varies from 3 to 17. As an example, here we briefly present the results for the, so called, Wang and Chua attractors.

The equations and the numerical setups for the experiments can be found in Appendix 3. Here we demonstrate again the impact of using the additional derivative information, as compared to the state values only.

For the Wang attractor [34] (13), examples of the velocity vector fields are shown in Figure 5a.

\[\begin{align*}
\dot{X} &= \alpha \cdot X + \gamma \cdot Y \cdot Z; \\
\dot{Y} &= \beta \cdot X + \delta \cdot Y - X \cdot Z; \\
\dot{Z} &= \epsilon \cdot Z + \zeta \cdot X \cdot Y.
\end{align*}\]

The 2D marginal parameter posteriors, obtained by a noiseless and noisy (with 5% Gaussian noise added again) data, respectively, are shown in Figure 5b. We see that the inclusion of the information on the dynamics dramatically shrinks the size of the parameter posterior distributions. Moreover, an inspection against the true underlying parameter values \((\alpha, \beta, \gamma, \delta, \epsilon, \zeta) = (0.2, -0.01, 1, -0.4, -1, -1)\) shows that the posteriors of the noisy case may be more biased if only the state values are
used, while the derivative information shrinks both the uncertainty and the bias of the posterior.

The Chua 7 attractor [39] (14), consists of seven main parts visible in Figure 6a. The equations of the system read as

\[
\begin{aligned}
\dot{X} &= a \cdot (Y - h); \\
\dot{Y} &= X - Y + Z; \\
\dot{Z} &= -\beta \cdot Y; \\
h &= m_7 \cdot X + 0.5 \\
&\times \left((m_0 - m_1) \cdot (|X + c_1| - |X - c_1|) + (m_1 - m_2) \cdot (|X + c_2| - |X - c_2|) + (m_2 - m_3) \cdot (|X + c_3| - |X - c_3|) + (m_3 - m_4) \cdot (|X + c_4| - |X - c_4|) + (m_4 - m_5) \cdot (|X + c_5| - |X - c_5|) + (m_5 - m_6) \cdot (|X + c_6| - |X - c_6|) + (m_6 - m_7) \cdot (|X + c_7| - |X - c_7|)\right).
\end{aligned}
\]  

(14)

In the parameter estimations both the state values and the velocities are used. The 2D marginal posteriors between the 17 parameters are shown in Figure 6b, obtained by a noiseless and noisy (with 1% relative Gaussian noise added) data, respectively. It is remarkable that all the 17 parameters can be identified by the CIL likelihood construction with high accuracy. The noiseless posteriors have maximally around 2% relative error, while the noisy measurements yield posteriors with maximum 6% relative error. A reason for such a high accuracy is partly due to the special character of the system: even moderately small steps outside the sampled parameter distributions may lead to qualitatively different attractors, with, e.g., some of the seven ‘discs’ missing. In the noiseless case the parameter posteriors show only mild correlations, while adding noise clearly increases the correlations between some of
the parameters. Most likely this is due to the high number of parameters and the largest errors added to the less frequently visited locations with the highest velocity values. We note (without details) that a CIL likelihood constructed with the same settings, but without the velocity values, also identifies all the parameters of the Chua 7 model, but with larger posterior distributions.

4. Discussion and conclusions. Estimation of the parameters of chaotic dynamical systems is complicated by the fact that any changes in the numerical solution, including solver settings for an otherwise fixed system, lead to different trajectories beyond an initial predictable time interval. In this sense no standard solution exists for the task of parameter estimation by long-time trajectory data. A partial solution is provided by filtering methods, which essentially split the total time interval into predictable subintervals and follow the trajectory of data by various prediction/correction methods. But these methods fail if the time intervals between data points are too long for predictions.

We present a new approach that is based on characterizing the set of trajectories corresponding to the given data, i.e., the underlying attractor, rather than trying to follow any specific trajectory close to the data. For this purpose we employ the distance concept introduced in [12], based on computing the statistics of the Ecdf of a point cloud of observations. A few ‘tuning’ parameters are needed for the likelihood construction, but they are usually found by a straightforward recipe.

With a large number of examples of chaotic systems we demonstrate how this approach allows one to perform the parameter estimation and subsequent MCMC sampling of the parameter posteriors. The initial parameter estimation is performed by the Differential Evolution method. It appears quite robust with respect to poor choice of the initial parameter values. Moreover, it often yields a remarkably good initial proposal for an adaptive MCMC sampling. No knowledge of initial values of the state vector are needed, as long as they stay in the domain of attraction.

Figure 6. Chua 7 model: parameters uncertainty quantification.
of data. Indeed, the initial states are randomized and an initial time interval of
simulations is discarded to guarantee that the point cloud used for the likelihood
construction samples the underlying attractor. For more complicated models, with
higher dimensions and consisting of coupled subsystems with different time scales,
more careful analysis is needed. We take a step in this direction by discussing an
approach that allows one to check whether a trajectory point has a ‘slow’ attractor
in its vicinity.

The presented approach may be extended in various ways. Here we develop
the inclusion of the time derivative of the state to the likelihood function. This,
even if approximated from noisy observations, greatly increases the accuracy of the
estimation results. Moreover, it allows the estimation of an overall time evolution
rate of the system. An obvious next step is to consider high dimensional chaotic
partial differential equation (PDE) systems. Challenges with increasing CPU times,
both in simulating the dynamical systems and in calculating distance matrices can
be treated in various ways: using parallel simulations and effective MCMC schemes,
such as early rejection, parallel sampling [32] and Local Approximation MCMC [5].

The presented approach can also, with natural modifications, be applied to prob-
lems other than chaotic systems, such as stochastic differential equations or deter-
ministic PDE systems that create random patterns from random initial values. The
basic idea there is the same as in the examples treated here: instead of trying to
predict an unpredictable solution, we characterize the variability of the outcomes,
and create a Gaussian likelihood that enables a robust statistical analysis of the
situation.

Acknowledgments. This work was supported by the Centre of Excellence of In-
verse Modelling and Imaging (CoE), Academy of Finland, decision number 312
122.

Author contributions statement. S.S. and H.H. worked out the dynamical ex-
tension of the approach. S.S. carried out the numerical studies. V.S. produced the
DE optimization. L.K. and D.S. performed the analysis of the multiple time scales.
All authors participated in writing the paper.

Appendix 1. The Appendix 1, contains the discussion about the stochastic opti-
mizer used in this work.

The DE algorithm requires the selection of mutation scheme and also allows
the use of additional techniques to enhance the diversity of the population to pre-
vent stagnation or convergence to a local minimum. All these parts of the algo-

rithm require several constants to be preliminarily fixed and the specific values
can be found in Table 1. Here we follow one of the standard mutation schemes,
“DE/current-to-best/1/bin” [8], which is enforced by the randomization strategy
of scale factor $F$ using the mixture of dither and jitter approaches [3]: $F_{i,g} = (F_{i}+\text{rand}_{g}(0,1))(F_{h}-F_{i})(1+\delta(\text{rand}_{j}(0,1)-0.5))$. This strategy allows the variabil-
ity of a specific value of a scale factor for each population member and its component
individually and improves exploratory quality of the algorithm. Also we utilize a
commonly used approach for keeping diversity of population, the opposition-based
generation jumping technique with elitist selection [28]. Another useful approach
particularly for stochastic environments is the recalculation step [31]. This step is
employed to entirely replace usual steps of the DE single evolution round at spec-
ified generations, and this helps to keep the statistics of the population goodness
up-to-date.
Table 1. DE settings used for all parameter estimations cases, see Appendix 3.

| Name                          | Notation | Value       |
|-------------------------------|----------|-------------|
| Crossover probability        | Cr       | 0.9         |
| Lower bound of scale factor  | FL       | 0.55        |
| Upper bound of scale factor  | FH       | 1.1         |
| Randomization factor         | δ        | 0.001       |
| Generation jumping probability | Jp      | 0.3         |
| Population size              | SoP      | 20 × number-of-parameters |

Appendix 2. For the readers interested in practical implementations of the described algorithms in Appendix 2, we preset the derivation of the formula for matrix $T$. We look for $T$ which satisfies following:

\[
T^{-1} J^0 T = T^{-1} \begin{pmatrix} F_0 & F_0 & F_0 \\ G_X & G_Y & G_Z \\ H_X & H_Y & H_Z \end{pmatrix} T = \begin{pmatrix} \lambda_1 & 0 & 0 & 0 \\ 0 & \lambda_2 & 0 & 0 \\ 0 & 0 & \lambda_3 & 0 \end{pmatrix} = \begin{pmatrix} \lambda_1 & 0 & 0 & 0 \\ 0 & \lambda_2 & 0 & 0 \\ 0 & 0 & \lambda_3 & 0 \end{pmatrix}
\]

Practically, we construct $T$ in the form:

\[
T = \begin{pmatrix} a_{11} & a_{21} & a_{31} & 0 \\ a_{12} & a_{22} & a_{32} & 0 \\ a_{13} & a_{23} & a_{33} & 0 \end{pmatrix},
\]

where $b_i, i = 1, 2, 3$ are the constants to be determined,

\[
P = (a_1, a_2, a_3) = \begin{pmatrix} a_{11} & a_{21} & a_{31} \\ a_{12} & a_{22} & a_{32} \\ a_{13} & a_{23} & a_{33} \end{pmatrix},
\]

such that

\[
P^{-1} f_x^0 P = \begin{pmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{pmatrix},
\]

and $a_1, a_2, a_3$ are the eigenvalues of $f_x^0$. The columns of $T$ defined in (16) are the eigenvectors of $J^0$. In particular, $(0, 0, 0, 1)^T$ is the eigenvector of $J^0$ corresponding to the eigenvalue $\lambda_4 = -K$. The constants $b_i, i = 1, 2, 3$ may be determined from the condition that the vectors $(a_{i1}, a_{i2}, a_{i3}, b_i)^T, i = 1, 2, 3$ must be the eigenvectors of $J^0$ corresponding to eigenvalues $\lambda_i, i = 1, 2, 3$. Satisfying this condition produces the following equations for $b_i, i = 1, 2, 3$:

\[
\begin{pmatrix} -F_0 & a_{i1} - F_0 & a_{i1} - F_0 & a_{i1} - K \end{pmatrix} = \lambda_i \begin{pmatrix} a_i \\ b_i \end{pmatrix}.
\]
For each $i = 1, 2, 3$ in (19) the first three equations are satisfied automatically; from the fourth equation,

\begin{equation}
-F_X^0 \cdot a_{i1} - F_Y^0 \cdot a_{i2} - F_Z^0 \cdot a_{i3} - K \cdot b_i = -(\partial f_i^0 / \partial x \cdot a_i) - K \cdot b_i = \lambda_i \cdot b_i,
\end{equation}

and thus,

\begin{equation}
b_i = \frac{-(\partial f_i^0 / \partial x \cdot a_i)}{K + \lambda_i},
\end{equation}

so that

\begin{equation}
T = \begin{pmatrix}
a_{11} & a_{21} & a_{31} & 0 \\
a_{12} & a_{22} & a_{32} & 0 \\
a_{13} & a_{23} & a_{33} & 0 \\
-(\partial f_i^0 / \partial x \cdot a_{1}) & -(\partial f_i^0 / \partial x \cdot a_{2}) & -(\partial f_i^0 / \partial x \cdot a_{3}) & 1
\end{pmatrix}
\end{equation}

\begin{equation}
= \begin{pmatrix}
P & 0 \\
0 & 0 \\
0 & 0 \\
0 & 0 & 1
\end{pmatrix}.
\end{equation}

It can be easily checked that

\begin{equation}
T^{-1} = \begin{pmatrix}
P^{-1} & 0 \\
0 & 0 \\
0 & 0 \\
0 & 0 & 1
\end{pmatrix},
\end{equation}

Appendix 3. In Appendix 3, we present all the attractors for which we performed the study, as well as corresponding choises of the parameter values.

Below $ft$ denotes the final time of integration, $N_o$ is the number of measurements and $N$ is the number of subsamples used to create the training sets.

(i) Lorenz63 [18]

\[
\begin{align*}
\dot{X} &= \beta \cdot (Y - X); \\
\dot{Y} &= X \cdot (\gamma - Z) - Y; \\
\dot{Z} &= X \cdot Y - \alpha \cdot Z.
\end{align*}
\]

$(\alpha, \beta, \gamma) = (8/3, 10, 28)$, $(X_0, Y_0, Z_0) = (0, 1.1, 20.2)$; $ft = 80000$, $N_o = 8000$, $N = 20000$, $M = M_{vel} = 10$, $R_0 = 2.66$, $b = 1.87$, $R_{0,vel} = 2.23$, $b_{vel} = 1.96$.

(ii) Rossler [19]

\[
\begin{align*}
\dot{X} &= -Y - Z; \\
\dot{Y} &= X + \alpha \cdot Y; \\
\dot{Z} &= \beta + Z \cdot (X - \gamma).
\end{align*}
\]

$(\alpha, \beta, \gamma) = (0.2, 0.2, 5.7)$, $(X_0, Y_0, Z_0) = (0.1, 0.1, 0.1);$ $ft = 400000$, $N_o = 8000$, $N = 20000$, $M = M_{vel} = 10$, $R_0 = 1.89$, $b = 1.81$, $R_{0,vel} = 2.07$, $b_{vel} = 1.87$. 
(iii) ChenLee [4]

\[
\begin{align*}
\dot{X} &= \alpha \cdot X - Y \cdot Z; \\
\dot{Y} &= \beta \cdot Y + X \cdot Z; \\
\dot{Z} &= \gamma \cdot Z + X \cdot Y/3.
\end{align*}
\]

\((\alpha, \beta, \gamma) = (5, -10, -0.38), (X_0, Y_0, Z_0) = (10; -5; -12); \)
\(ft = 160000, \ N_0 = 8000, \ N = 2000, \)
\(M = M_{vel} = 10, \ R_0 = 2.54, \ b = 2.24, \ R_{0,vel} = 2.47, \ b_{vel} = 2.69.\)

(iv) Genesio [9]

\[
\begin{align*}
\dot{X} &= Y; \\
\dot{Y} &= Z; \\
\dot{Z} &= \gamma \cdot X - \beta \cdot Y - \alpha \cdot Z + X^2.
\end{align*}
\]

\((\alpha, \beta, \gamma) = (0.44, 1.1, 1), (X_0, Y_0, Z_0) = (0.1, 0.1, 0.1); \)
\(ft = 240000, \ N_0 = 8000, \ N = 2000, \)
\(M = M_{vel} = 10, \ R_0 = 2.25, \ b = 1.82, \ R_{0,vel} = 2.26, \ b_{vel} = 1.82.\)

(v) LuChen [16]

\[
\begin{align*}
\dot{X} &= -(\alpha \cdot \beta \cdot X)/((\alpha + \beta) - Y \cdot Z + \gamma); \\
\dot{Y} &= \alpha \cdot Y + X \cdot Z; \\
\dot{Z} &= \beta \cdot Z + X \cdot Y.
\end{align*}
\]

\((\alpha, \beta, \gamma) = (-10, -4.18.2), (X_0, Y_0, Z_0) = (10, -5, -12); \)
\(ft = 160000, \ N_0 = 8000, \ N = 2000, \)
\(M = M_{vel} = 10, \ R_0 = 1.77, \ b = 1.97, \ R_{0,vel} = 2.42, \ b_{vel} = 2.30.\)

(vi) Chua3 [20]

\[
\begin{align*}
\dot{X} &= \alpha \cdot (Y - X - h); \\
\dot{Y} &= X - Y + Z; \\
\dot{Z} &= -\beta \cdot Y; \\
\dot{h} &= m_1 \cdot X + 0.5 \cdot (m_0 - m_1) \cdot (|X + 1| - |X - 1|).
\end{align*}
\]

\((\alpha, \beta, m_0, m_1) = (15.6, 28, -8/7, -5/7), (X_0, Y_0, Z_0) = (0.1, 0.1, 0.1); \)
\(ft = 160000, \ N_0 = 8000, \ N = 2000, \)
\(M = M_{vel} = 10, \ R_0 = 2.85, \ b = 1.97, \ R_{0,vel} = 2.59, \ b_{vel} = 1.77.\)

(vii) Lorenz2 [23]

\[
\begin{align*}
\dot{X} &= -\alpha \cdot X + Y^2 - Z^2 + \alpha \cdot \gamma; \\
\dot{Y} &= X \cdot (Y - \beta \cdot Z) + \delta; \\
\dot{Z} &= -Z + X \cdot (\beta \cdot Y + Z).
\end{align*}
\]

\((\alpha, \beta, \gamma, \delta) = (0.9, 5, 9.9, 1), (X_0, Y_0, Z_0) = (0.1, 0.1, 0.1); \)
\(ft = 80000, \ N_0 = 8000, \ N = 2000, \)
\(M = M_{vel} = 10, \ R_0 = 2.56, \ b = 1.73, \ R_{0,vel} = 2.11, \ b_{vel} = 2.00.\)
(viii) Yu Wang [21]

\[
\begin{align*}
\dot{X} & = \alpha \cdot (Y - X); \\
\dot{Y} & = \beta \cdot X + \gamma \cdot X \cdot Z; \\
\dot{Z} & = \exp(X \cdot Y) - \delta \cdot Z.
\end{align*}
\]

\((\alpha, \beta, \gamma, \delta) = (10, 40, 2, 2.5), (X_0, Y_0, Z_0) = (0.1, 0.1, 0.1);\)

\(ft = 320000, \ N_0 = 8000, \ N = 2000;\)

\(M = M_{vel} = 10, \ R_0 = 2.62, \ b = 1.86, \ R_{0,vel} = 1.87, \ b_{vel} = 2.04.\)

(ix) TSUCS1 [25]

\[
\begin{align*}
\dot{X} & = \alpha \cdot (Y - X) + \gamma \cdot X \cdot Z; \\
\dot{Y} & = \epsilon \cdot Y - X \cdot Z; \\
\dot{Z} & = \beta \cdot Z + X \cdot Y - \delta \cdot X^2.
\end{align*}
\]

\((\alpha, \beta, \gamma, \delta, \epsilon) = (40, 1.8333, 0.5, 0.65, 20), (X_0, Y_0, Z_0) = (0.1, 0.1, 0.1);\)

\(ft = 160000, \ N_0 = 8000, \ N = 2000;\)

\(M = M_{vel} = 10, \ R_0 = 2.39, \ b = 1.91, \ R_{0,vel} = 2.36, \ b_{vel} = 2.12.\)

(x) TSUCS2 [25]

\[
\begin{align*}
\dot{X} & = \alpha \cdot (Y - X) + \delta \cdot X \cdot Z; \\
\dot{Y} & = \beta \cdot X - X \cdot Z + \zeta \cdot Y; \\
\dot{Z} & = \gamma \cdot Z + X \cdot Y - \epsilon \cdot X^2.
\end{align*}
\]

\((\alpha, \beta, \gamma, \delta, \epsilon, \zeta) = (40, 55, 1.833, 16, 0.65, 20), (X_0, Y_0, Z_0) = (10, 10, 10);\)

\(ft = 420000, \ N_0 = 8000, \ N = 2000;\)

\(M = M_{vel} = 10, \ R_0 = 2.40, \ b = 1.74, \ R_{0,vel} = 2.39, \ b_{vel} = 2.02.\)

(xi) Aizawa [6]

\[
\begin{align*}
\dot{X} & = (Z - \epsilon) \cdot X - \delta \cdot Y; \\
\dot{Y} & = \delta \cdot X + (Z - \epsilon) \cdot Y; \\
\dot{Z} & = \gamma + \beta \cdot Z - Z^3/3 \\
& \quad - (X^2 + Y^2) \cdot (1 + \alpha \cdot Z) + \zeta \cdot Z \cdot X^3.
\end{align*}
\]

\((\alpha, \beta, \gamma, \delta, \epsilon, \zeta) = (0.25, 0.95, 0.6, 3.5, 0.7, 0.1), (X_0, Y_0, Z_0) = (0.1, 0.1, 0.1);\)

\(ft = 320000, \ N_0 = 8000, \ N = 2000;\)

\(M = M_{vel} = 10, \ R_0 = 2.01, \ b = 1.71, \ R_{0,vel} = 2.02, \ b_{vel} = 1.73.\)

(xii) Wang [34]

\[
\begin{align*}
\dot{X} & = \alpha \cdot X + \gamma \cdot Y \cdot Z; \\
\dot{Y} & = \beta \cdot X + \delta \cdot Y - X \cdot Z; \\
\dot{Z} & = \epsilon \cdot Z + \zeta \cdot X \cdot Y.
\end{align*}
\]

\((\alpha, \beta, \gamma, \delta, \epsilon, \zeta) = (0.2, -0.01, 1, -0.4, -1, -1), (X_0, Y_0, Z_0) = (1, 1, 1);\)

\(ft = 120000, \ N_0 = 32000, \ N = 8000;\)

\(M = M_{vel} = 10, \ R_0 = 2.17, \ b = 2.16, \ R_{0,vel} = 1.23, \ b_{vel} = 2.61.\)
(xiii) Chua5 [38]
\[
\begin{align*}
\dot{X} &= \alpha \cdot (Y - h) \\
\dot{Y} &= X - Y + Z \\
\dot{Z} &= -\beta \cdot Y \\
h &= m_5 \cdot X + 0.5 \\
&\quad \times \left( (m_0 - m_1) \cdot (|X + c_1| - |X - c_1|) \\
&\quad + (m_1 - m_2) \cdot (|X + c_2| - |X - c_2|) \\
&\quad + (m_2 - m_3) \cdot (|X + c_3| - |X - c_3|) \\
&\quad + (m_3 - m_4) \cdot (|X + c_4| - |X - c_4|) \\
&\quad + (m_4 - m_5) \cdot (|X + c_5| - |X - c_5|) \right).
\end{align*}
\]

\( (\alpha, \beta, m_0, m_1, m_2, m_3, m_4, m_5, c_1, c_2, c_3, c_4, c_5) = (14, 20, 0.9/7, -3/7, 3.5/7, -2/7, 4/7, -2.4/7, 1.0, 2.15, 3.6, 6.2, 9), \)

\( (X_0, Y_0, Z_0) = (0.1, 0.1, 0.1); \)

\( ft = 240000, N_0 = 16000, N = 4000, \)

\( M = M_{vel} = 10, R_0 = 3.03, b = 1.86, R_{0, vel} = 2.50, b_{vel} = 1.79. \)

(xiv) Chua7 [39]
\[
\begin{align*}
\dot{X} &= \alpha \cdot (Y - h) \\
\dot{Y} &= X - Y + Z \\
\dot{Z} &= -\beta \cdot Y \\
h &= m_7 \cdot X + 0.5 \\
&\quad \times \left( (m_0 - m_1) \cdot (|X + c_1| - |X - c_1|) \\
&\quad + (m_1 - m_2) \cdot (|X + c_2| - |X - c_2|) \\
&\quad + (m_2 - m_3) \cdot (|X + c_3| - |X - c_3|) \\
&\quad + (m_3 - m_4) \cdot (|X + c_4| - |X - c_4|) \\
&\quad + (m_4 - m_5) \cdot (|X + c_5| - |X - c_5|) \\
&\quad + (m_5 - m_6) \cdot (|X + c_6| - |X - c_6|) \\
&\quad + (m_6 - m_7) \cdot (|X + c_7| - |X - c_7|) \right).
\end{align*}
\]

\( (\alpha, \beta, m_0, m_1, m_2, m_3, m_4, m_5, m_6, m_7, c_1, c_2, c_3, c_4, c_5, c_6, c_7) = (14, 20, 0.9/7, -3/7, 0.5, -0.3429, 0.36, -0.24, 0.36, -0.24, -0.3429, 1, 2.15, 6.2, 9, 14.25), \)

\( (X_0, Y_0, Z_0) = (0.1, 0.1, 0.1); \)

\( ft = 240000, N_0 = 16000, N = 4000, \)

\( M = M_{vel} = 10, R_0 = 2.9, b = 1.88, R_{0, vel} = 2.49, b_{vel} = 1.82. \)

REFERENCES

[1] M. A. Beaumont, W. Y. Zhang and D. J. Balding, Approximate bayesian computation in population genetics, Genetics, 162 (2002), 2025–2035.

[2] S. Borovkov, R. Burton and H. Dehling, Limit theorems for functionals of mixing processes with applications to U-statistics and dimension estimation, Transactions of the American Mathematical Society, 353 (2001), 4261–4318.

[3] U. K. Chakraborty, Computational intelligence, Springer: Verlag, 143 (2008), 338.

[4] H.-K. Chen and C.-I. Lee, Anti-control of chaos in rigid body motion, Chaos, Solitons and Fractals, 21 (2004), 957–965.

[5] P. R. Conrad, Y. M. Marzouk, N. S. Pillai and A. Smith, Accelerating asymptotically exact MCMC for computationally intensive models via local approximations, Journal of the American Statistical Association, 111 (2016), 1591–1607.

[6] G. Coper, Aizawa strange attractor, (2018), http://www.algosome.com/articles/aizawa-attractor-chaos.html.
Robust parameter estimation of chaotic systems

[7] J. Durbin and S. J. Koopman, *Time Series Analysis by State Space Methods*, Oxford Statistical Science Series, 38, Oxford University Press, Oxford, 2012.

[8] V. Feoktistov, *Differential Evolution: In Search of Solutions*, Springer Optimization and Its Applications, 5, Springer, New York, 2006.

[9] R. Genesio and A. Tesi, "Harmonic balance methods for the analysis of chaotic dynamics in nonlinear systems," *Automatica*, 28 (1992), 531–548.

[10] H. Haario, E. Saksman and J. Tamminen, "An adaptive Metropolis algorithm," *Bernoulli Society for Mathematical Statistics and Probability*, 7 (2001), 223–242.

[11] H. Haario, M. Laine, A. Mira and E. Saksman, "DRAM: Efficient adaptive MCMC," *Statistics and Computing*, 16 (2006), 339–354.

[12] H. Haario, L. Kalachev and J. Hakkarainen, "Generalized correlation integral vectors: A distance concept for chaotic dynamical systems," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 25 (2015), Article 063102, 10 pp.

[13] J. Hakkarainen, A. Ilin, A. Solonen, M. Laine, H. Haario, J. Tamminen, E. Oja and H. Järvinen, "On closure parameter estimation in chaotic systems," *Nonlinear Processes in Geophysics*, 19 (2012), 127–143.

[14] J. Hakkarainen, A. Solonen, A. Ilin, J. Susiluoto, M. Laine, H. Haario and H. Järvinen, "A dilemma of the uniqueness of weather and climate model closure parameters," *Tellus A: Dynamic Meteorology and Oceanography*, 65 (2013), 20147.

[15] H. Järvinen, P. Räisänen, M. Laine, J. Tamminen, A. Ilin, E. Oja, A. Solonen and H. Haario, "Estimation of ECHAM5 climate model closure parameters with adaptive MCMC," *Atmospheric Chemistry and Physics*, 10 (2010), 9993–10002.

[16] J. H. Lü, G. R. Chen and S. C. Zhang, "Dynamical analysis of a new chaotic attractor," *International Journal of Bifurcation and Chaos*, 12 (2002), 1001–1015.

[17] E. Kalnay, *Atmospheric Modeling, Data Assimilation and Predictability*, Cambridge University Press, 2002.

[18] E. N. Lorenz, "Deterministic nonperiodic flow," *Journal of the Atmospheric Sciences*, 20 (1963), 130–141.

[19] E. N. Lorenz, "An equation for continuous chaos," *Physics Letters A*, 57 (1976), 397–398.

[20] T. Matsumoto, "A chaotic attractor from Chua's circuit," *IEEE Transactions on Circuits and Systems*, 31 (1984), 1055–1058.

[21] T. Matsumoto, "Generation of a new three dimension autonomous chaotic attractor and its four wing type," *ETASR Engineering, Technology and Applied Science Research*, 3 (2013), 352–358.

[22] D. McFadden, "A method of simulated moments for estimation of discrete response models without numerical integration," *Econometrica*, 57 (1989), 995–1026.

[23] J. Meier, "Lorenz Mod 2 strange attractor," (2018), http://3d-meier.de/tut19/Seite80.html.

[24] S. Rahnamayan, H. R. Tizhoosh and M. M. A. Salama, "Opposition-based differential evolution," *Advances in Differential Evolution*, (2008), 155–171.

[25] J. Rougier, "Intractable and unsolved": Some thoughts on statistical data assimilation with uncertain static parameters, *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 371 (2013), 20120297, 12 pp.

[26] V. Shemyakin and H. Haario, "Online identification of large scale chaotic system," *Nonlinear Dynamics*, 93 (2018), 961–975.

[27] A. Solonen, P. Ollinaho, M. Laine, H. Haario, J. Tamminen and H. Järvinen, "Asymmetric approach to generating n-scroll attractor," *International Society for Bayesian Analysis*, 12 (2012), 715–736.
[33] R. Storn and K. Price, Differential evolution: A simple and efficient heuristic for global optimization over continuous spaces, *Journal of Global Optimization*, 11 (1997), 341–359.

[34] Z. H. Wang, Y. X. Sun, B. J. van Wyk, G. Y. Qi and M. A. van Wyk, A 3-D four-wing attractor and its analysis, *Brazilian Journal of Physics*, 39 (2009), 547–553.

[35] W.-H. Ho, J.-H. Chou and C.-Y. Guo, Parameter identification of chaotic systems using improved differential evolution algorithm, *Nonlinear Dynamics*, 61 (2010), 29–41.

[36] S. N. Wood, Statistical inference for noisy nonlinear ecological dynamic systems, *Nature*, 466 (2010), 1102–1104.

[37] X. T. Li and M. H. Yin, Parameter estimation for chaotic systems by hybrid differential evolution algorithm and artificial bee colony algorithm, *Nonlinear Dynamics*, 77 (2014), 61–71.

[38] M. E. Yalcin, J. A. K. Suykens and J. Vandewalle, Experimental confirmation of 3- and 5-scroll attractors from a generalized Chua’s circuit, *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 47 (2000), 425–429.

[39] G.-Q. Zhong, K.-F. Man and G. R. Chen, A systematic approach to generating n-scroll attractors, *International Journal of Bifurcation and Chaos*, 12 (2002), 2907–2915.