Journal of Physics:
Complexity

PURPOSE-LED
PUBLISHING™

PAPER • OPEN ACCESS

A generalized method for estimating parameters of chaotic systems using synchronization with modern optimizers

To cite this article: Davide Prosperino *et al* 2025 *J. Phys. Complex.* **6** 015012

View the article online for updates and enhancements.

You may also like

- Zoo guide to network embedding
A Baptista, R J Sánchez-García, A Baudot et al.
- Biological arrow of time: emergence of tangled information hierarchies and self-modelling dynamics
Mikhail Prokopenko, Paul C W Davies, Michael Harré et al.
- Sensitivity to network perturbations in the randomized shortest paths framework: theory and applications in ecological connectivity
Ilkka Kivimäki, Bram Van Moorter and Marco Saerens

This content was downloaded from IP address 129.247.247.239 on 10/03/2025 at 10:14

IOP Publishing J. Phys. Complex. 6 (2025) 015012 https://doi.org/10.1088/2632-072X/adaa46

Journal of Physics: Complexity

OPEN ACCESS

RECEIVED
19 February 2024

REVISED
7 January 2025

ACCEPTED FOR PUBLICATION
14 January 2025

PUBLISHED
3 March 2025

Original Content from this work may be used under the terms of the Creative Commons Attribution 4.0 licence.

Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.

PAPER

A generalized method for estimating parameters of chaotic systems using synchronization with modern optimizers

Davide Prosperino[1], **Haochun Ma**[1] and **Christoph Räth**[2,*]

[1] Faculty of Physics, Ludwig-Maximilians-Universität München, Geschwister-Scholl-Platz 1, 80539 Munich, Germany
[2] Institute of Materials Physics in Space, Deutsches Zentrum für Luft- und Raumfahrt (DLR), Linder Höhe, 51147 Cologne, Germany
* Author to whom any correspondence should be addressed.

E-mail: christoph.raeth@dlr.de

Keywords: synchronization, parameter estimation, Lorenz system, coupling

Abstract

Deriving governing equations from time series data is an ongoing topic of research across different disciplines in science. While the terms of the governing equations can be reconstructed by combinations of the input coordinates or other more sophisticated methods, inferring the coefficients of each term is a complex task on its own. Here, we extend and discuss an algorithm for finding the correct coefficients of the governing equations of chaotic systems by introducing a unidirectional coupling. We achieve this by treating the data as a primary system and coupling a secondary system to it. Then by inducing synchronization, we can push the parameters of the secondary system in the direction minimizing a loss function. After the loss has reached its minimum, the found parameters are a good estimate of the real parameters producing the data. We apply our algorithm on numerous chaotic systems and we find that this method identifies the correct coefficients for all of them, while being robust to noise and incorrect terms in the governing equations. Additionally, we discover that the Lorenz equations are not the only ones producing the—or *a*—butterfly-shaped attractor.

1. Introduction

With the explosion of computational resources more and more data is produced and processed than ever before. While various techniques exist to use this data and create models & forecasts from it, most modern approaches hide their functionality in a high-dimensional space and can therefore be seen as 'black box'-approaches. On such approach commonly used in physics being reservoir computing [1].

In contrast to that, one approach consists of deducing the governing equations from a given time series. This has the advantage of rendering the obtained model understandable and explainable. Such models could be used in the modeling of the climate [2, 3] and turbulent air flows [4], epidemics [5, 6], and even in financial markets [7–9]; all fields, where explainability is as important as accuracy.

Many different models for finding governing equations from data emerged in recent years [9–12]. Most methods, such as sparse identification of nonlinear dynamics (SINDy) [11], perform regressions on a large set of terms, which are constructed from all possible combinations of the input variables. However, for high-dimensional and nonlinear data the number of possible terms explodes. Therefore, Ma *et al* [9] introduced a method using causal inference to identify only relevant terms. Now, the task remains to calibrate a set of dynamical equations by finding the coefficients which describe the data best.

Current work on this topic exists and can be categorized by their different approaches: a genetic algorithm can by found by Tao *et al* [13], and a particle swarm optimization can be found by Mukhopadhyay and Banerjee [14]. In [15] we found a useful list of possible different approaches to this problem.

Another prominent approach entails utilizing synchronization, which can be obtained by coupling two systems. While Bagnoli and Baia [16] study the coupling in the parameters of dynamical systems, a more traditional approach consists of coupling the system's coordinates. This phenomenon has been described by Fujisaka and Yamada [17] and Pecora and Carroll [18]. Following work by Parlitz *et al*[19, 20] used this

© 2025 The Author(s). Published by IOP Publishing Ltd

synchronization to estimate parameters from time series. However, in their work they relied on a Lorenz63-specific coupling. Similarly, other work on this area so far also relied on non-trivial coupling methods, which are model specific and need to be determined beforehand for the system at hand [21–25]. In section 3.1 we will make the synchronization independent of the system and thus use a coupling, which works for all chaotic systems and does not require prior calculations. This is especially important in scenarios where prior calculations are not feasible or the prior calculations are not guaranteed to be correct, due to uncertainties in the structure of the governing equations.

Work by Abarbanel *et al* [26] puts this topic in the context of data assimilation, however, it lacks a detailed analysis on the synchronization and stability considerations. While work by Maybhate and Amritkar [21] and by Creveling *et al* [27] apply these ideas on an incomplete data setup, in this work we focus on situations where all data is available and observable.

Typically, the Lyapunov function is used to show stability and convergence of a coupling [22–24]. However, Mariño and Míguez [28] used the mean-squared-error due to its general applicability and simplicity to derive a gradient from it. In this article we also focus on gradient descent-based algorithms, since the recent advent of deep-learning technologies has led to a family of improved first-order optimizers, such as 'Adam' [29] or 'AMSGrad' [30]. An algorithm based on first-order optimizations was introduced by Mariño and Míguez [28]. However, their work is limited to the Lorenz63 system and lacks general applicability.

The main idea of this work is to generalize the findings of Mariño and Míguez [28] by extending their algorithm and making it applicable to arbitrary chaotic systems. Furthermore, we modernize it by extending it with advances in stochastic optimization. Regarding the theoretical aspect, we will demonstrate that our algorithm indeed utilizes the synchronization property of chaotic systems. We will highlight that our modifications make this algorithm more robust against noise and incorrect knowledge of the governing equations compared to the original benchmark in [28]. Additionally, we will compare it to the SINDy algorithm in [11].

In the end we will apply our algorithm on different libraries of terms and try to reconstruct the Lorenz63 equations. There we will find that different sets of equations exist, which produce an attractor with the iconic butterfly shape.

2. Data

To prove its general applicability, we apply our algorithm on a number of chaotic systems, as we will show in section 4. However, as an example during this paper we will introduce and visualize our ideas using the Lorenz63 system [2] described by

$$\dot{x}_1 = -\sigma x_1 + \sigma x_2 \tag{1a}$$

$$\dot{x}_2 = \rho x_1 - x_2 - x_1 x_3 \tag{1b}$$

$$\dot{x}_3 = -\beta x_3 + x_1 x_2 \quad . \tag{1c}$$

We use the standard parametrization of $\sigma = 10$, $\rho = 28$, and $\beta = \frac{8}{3}$, and store the parameters in a parameter vector $\underline{\theta} = (\sigma, \rho, \beta)^\intercal$. In order to sample a trajectory, we use three random integers ranging from -20 to 20 as initial condition and integrate the system using the LSODA algorithm introduced by Hindmarsh and Petzold [31] with a step size of $\Delta t = 10^{-2}$, unless stated otherwise. Afterwards, we discard the first 10^3 data points as transient behavior until we converge onto the attractor.

3. Algorithm

For our algorithm we follow the basic idea introduced by Mariño and Míguez [28]. We start with two systems: a primary one, and a secondary one. The primary system is the measured data and consists of discrete, observed data points. The secondary system encodes our prior knowledge of the structure of the governing equations, and an initial guess of the parameters to estimate.

Figure 1 shows a schematic overview of our algorithm. The primary system, represented by $\underline{x}$, is created by unknown parameters with a known structure of governing equations F. From that, discrete samples are observed. The secondary system, $\underline{y}$, is calculated by using guessed parameters and the structure F. The secondary system is coupled unidirectionally to the primary one. The unidirectional coupling is required since the observed data points are immutable. We then calculate a loss function between the secondary system and the primary one. Since the computation rule of the loss is known, we can calculate the gradient which minimizes the loss function. We take a step in the minimizing direction and repeat.

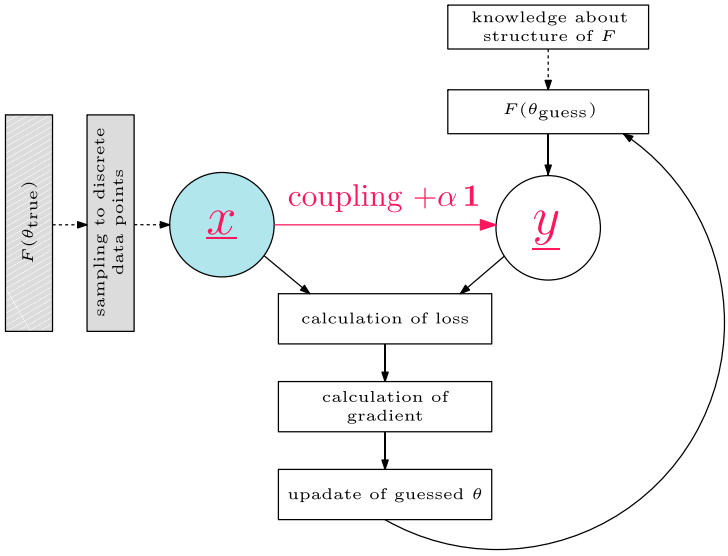


Figure 1. In this schematic representation of our algorithm the gray boxes represent knowledge not known at time of the optimization. The blue box indicates the immutability of the observed data $\underline{x}$. The dotted line represents the initialization step and the solid lines represent the steps taken at each iteration step.

Once the loss converges to a minimal value, we know that the guessed parameters are a good approximation for the system producing the primary data points.

In the following sections we will dissect our algorithm and introduce the details of each part. Section 3.1 discusses a general coupling which works for all systems and leads to guaranteed synchronization. In section 3.2 we emphasize the synchronization aspect using the Lorenz63 system as an example. In section 3.3 we introduce our loss function and in section 3.4 the calculation of its gradient. Section 3.5 presents the modern optimizers which we use to calculate the concrete steps of the guessed parameters.

3.1. Synchronization

In their work, Eroglu *et al* [32] presented a way to couple two arbitrary chaotic systems of the same kind to each other. Meaning that the trajectories of each systems will converge to a common one. In this work, we extend their argument to the case, where we synchronize a secondary system *to* a primary one. This leads to the trajectory of the primary system to remain unchanged, while the secondary system's trajectory gets pushed to behave like the primary one. For our case we require the trajectory of the primary system to remain unchanged, since one application is to use discrete data points as a primary system, which cannot be changed.

We define two arbitrary N-dimensional systems $\underline{x}, \underline{y} \in \mathbb{R}^N$. Inspired by the method of Eroglu *et al* [32], we add the coupling $\alpha \mathbf{H}\left(\underline{x} - \underline{y}\right)$ to the system $\underline{y}$ with $\mathbf{H} = \mathbf{1}$, the identity matrix. This makes $\underline{x}$ the primary, driving system and $\underline{y}$ the secondary, driven system. This leads to the following dynamical description:

$$\dot{\underline{x}} = F(\underline{x}) \tag{2a}$$

$$\dot{\underline{y}} = F\left(\underline{y}\right) + \alpha \mathbf{H}\left(\underline{x} - \underline{y}\right) \quad . \tag{2b}$$

Here, F: $\mathbb{R}^N \to \mathbb{R}^N$ describes the evolution of the dynamical system and the matrix $\mathbf{H}$: $\mathbb{R}^N \to \mathbb{R}^N$ describes the coupling of the secondary system to the primary one. We require $\mathbf{H}(\underline{0}) = \underline{0}$, meaning that for synchronized systems the coupling vanishes as soon as both systems are on the same trajectory [32]. The parameter $\alpha \in \mathbb{R}^+$ is named 'coupling strength'.

Following the argumentation by Eroglu *et al* [32], we will show that if the coupling strength is sufficiently large, the systems described by equations (2) will synchronize. For the rest of this argument we consider the identity coupling, meaning $\mathbf{H} = \mathbf{1}$. This simplifies the coupling term in equation (2*b*) to $\alpha \mathbf{H}\left(\underline{x} - \underline{y}\right) = \alpha \left(\underline{x} - \underline{y}\right)$. Additionally, we define a difference variable $\underline{z} = \underline{x} - \underline{y}$. Using equations (2), we can describe the evolution of the new variable $\underline{z}$ by

$$\dot{\underline{z}} = \dot{\underline{x}} - \dot{\underline{y}} = F(\underline{x}) - F\left(\underline{y}\right) - \alpha \underline{z} \quad . \tag{3}$$

Now synchronization can be defined as the limit $\lim_{t\to\infty} \underline{z} = \underline{0}$. Since we are interested in the synchronized state, meaning $\underline{z} = \underline{x} - \underline{y} = \underline{0}$, we can Taylor-expand the system $F\left(\underline{y}\right)$ at said criterion $\underline{x} = \underline{y}$ leading to

$$\begin{aligned} F\left(\underline{y}\right) &= F(\underline{x}) + \mathrm{D}F(\underline{x})\,\left(\underline{y} - \underline{x}\right) + \mathcal{O}\left(||\underline{y} - \underline{x}||^2\right) \\ &= F(\underline{x}) - \mathrm{D}F(\underline{x})\,\underline{z} + \mathcal{O}\left(||\underline{z}||^2\right) \quad . \end{aligned} \tag{4}$$

$\mathrm{D}F(\underline{x})$ describes the Jacobian matrix of $F(\underline{x})$. Plugging in the Taylor-expansion in equation (3) leads to the following first-order approximation at the synchronization criterion for $\underline{z}$:

$$\dot{\underline{z}} = \left(\mathrm{D}F(\underline{x}) - \alpha\,\mathbf{1}\right)\,\underline{z} \quad . \tag{5}$$

Equation (5) is a non-autonomous equation, since it depends explicitly on the solution $\underline{x}$, so its analysis is non-trivial. We introduce the variable $\underline{w} = \exp\left(\alpha\,t\right)\underline{z}$ in order to get rid of the term $-\alpha\,\mathbf{1}\,\underline{z}$ in equation (5). The derivative of the new variable $\underline{w}$ is

$$\dot{\underline{w}} = \alpha\,\exp\left(\alpha\,t\right)\underline{z} + \exp\left(\alpha\,t\right)\dot{\underline{z}} \quad . \tag{6}$$

Using the definition of $\underline{w}$ and its derivative in equation (6), equation (5) simplifies to

$$\dot{\underline{w}} = \left(\mathrm{D}F(\underline{x})\right)\,\underline{w} \quad . \tag{7}$$

Equation (7) is the first variational equation for $\underline{x}$ solving $\dot{\underline{\xi}} = F\left(\underline{\xi}\right)$.

Now, let $\mathbf{\Phi}$ be the fundamental matrix for equation (7). This matrix is a function of $\underline{x}$. By definition, this means that any solution of $\underline{w}$ can be written as $\underline{w} = \mathbf{\Phi}\underline{w}(\underline{0})$ for any initial condition $\underline{w}(\underline{0})$, and by substitution the same can be done for $\underline{z}$. Next, we consider the matrix $\mathbf{\Phi}^{\mathsf{T}}\mathbf{\Phi}$. The square root of the largest eigenvalue of the matrix $\mathbf{\Phi}^{\mathsf{T}}\mathbf{\Phi}$ is the largest Lyapunov exponent [33]. Therefore, if λ denotes the largest Lyapunov exponent of $\underline{x}$, the following inequality holds for the difference variable $\underline{w}$ [32]:

$$||\underline{w}|| \leqslant C\exp\left(\lambda\,t\right) \quad . \tag{8}$$

Resubstituting the definition of $\underline{z}$ into equation (8) yields the following inequality:

$$||\underline{z}|| \leqslant C\exp\left(\left(\lambda - \alpha\right)\,t\right) \quad . \tag{9}$$

Reminding that $\underline{z}$ describes the difference between the two systems $\underline{x}$ and $\underline{y}$, we see that if the exponent in equation (9) is negative, the difference between the systems $\underline{x}$ and $\underline{y}$ will tend to 0 as time progresses. Therefore, we define a critical coupling α_{c} so that the exponent is negative with

$$\alpha_{\mathrm{c}} = \lambda \quad . \tag{10}$$

If the coupling strength α is bigger than α_{c}, the largest Lyapunov exponent, the two systems will synchronize. This finding is visualized in figure 2 for the Lorenz63 system, where we can clearly observe a sharp drop in the synchronization loss between the two systems. For that, we define the synchronization loss E_{s} as the average deviation from synchronization between the two systems with

$$E_{\mathrm{s}} = \frac{1}{N}\sum_{i=1}^{N} ||\underline{x}_i - \underline{y}_i|| \quad . \tag{11}$$

This definition of the synchronization loss includes the whole history of both systems' trajectories, meaning that it also includes the unsynchronized phase at the beginning. For this reason the synchronization loss in figure 2 does not drop to 0, but instead encapsulates the time needed for synchronization.

Using this coupling, we can synchronize two arbitrary chaotic systems of the same kind to each other while maintaining the notion of a driving system and a driven system.

While this coupling method guarantees a unidirectional synchronization, equation (9) has one caveat: due to the constant C it is not possible to estimate a time t, after which the distance between the two systems $||\underline{z}||$ is sufficiently small. Meaning that, while we can guarantee synchronization eventually, we cannot make a statement on how quickly the synchronized state will be reached. However, experimentally, we find that with an increasing coupling strength α, we are able to reduce the time until synchronization. This finding is intuitive when analyzing equation (9), since the higher the magnitude of the exponent, the faster the distance $||\underline{z}||$ decays.

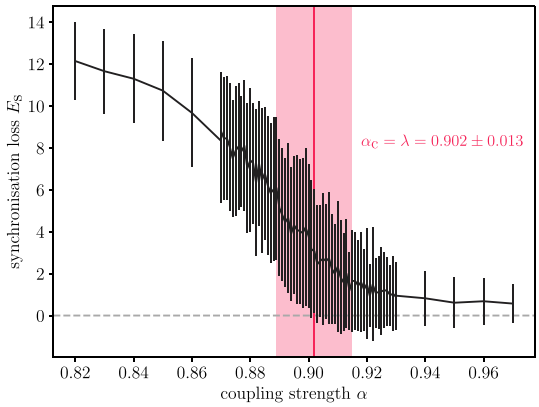


Figure 2. We observe a sharp drop of the synchronization E_s loss comparing the regimes of weak coupling ($\alpha < \lambda$) and strong coupling ($\alpha > \lambda$). We used the Lorenz63 system as defined equations (1). Using the algorithm described in [34], we get $\lambda = 0.902 \pm 0.013$ as the largest Lyapunov exponent for the Lorenz63 system. This figure shows the mean and standard deviation for a hundred iterations for each coupling strength.

While Creveling *et al* [27] found an upper limit for the coupling strength in their setup with incomplete observations, our setup does not inhibit such an upper boundary, as equation (9) suggests that a higher coupling leads to a faster synchronization. We validated this finding numerically and found correct convergence of the parameters for coupling strengths up to 10^9.

This means for our optimization algorithm that we can set the coupling strength α to a comparably high value, since it will only affect the time to synchronization.

3.2. Linear stability analysis

So far, we assumed in equation (2) that the governing equations for both systems are identical in term structure and parameters. However, in our definition of the problem we only know the structure of the equations and still need an initial guess for the parameters. For this case we need to alter the dynamical description to

$$\dot{\underline{x}} = F(\underline{x})$$
$$\dot{\underline{y}} = G\left(\underline{y}\right) + \alpha\left(\underline{x} - \underline{y}\right) \tag{12a}$$
$$= F\left(\underline{y}\right) + E\left(\underline{y}\right) + \alpha\left(\underline{x} - \underline{y}\right) \quad , \tag{12b}$$

where we already assumed a coupling using the identity matrix $\mathbf{H} = \mathbf{1}$. Here, we decompose the secondary system G with the wrong parameters into a system F with the correct parameters and an error system E, encoding our wrong initial knowledge of the parameters. Systems F and E (and therefore G too) have the same structure but different parameters for the terms.

We analyze the synchronization property by studying the linearized dynamics of perturbations around the synchronized state. The general idea of this approach was introduced by Pecora and Carroll [35], and in this article we perform a specialized analysis of it.

We denote the synchronized state with $\underline{s}$ and look at the errors $\underline{\eta}$ for each system, the driving and the driven one. We arrive at

$$\underline{\eta}_x = \underline{x} - \underline{s}$$
$$\underline{\eta}_y = \underline{y} - \underline{s} \quad .$$

We Taylor-expand the evolution up to first order at the synchronized state $\underline{s} = \underline{x} = \underline{y}$, which leads to the approximate evolution of the perturbations $\underline{\eta}$ described by

$$\dot{\underline{\eta}}_x = \mathrm{D}F(s)\,\underline{\eta}_x \tag{13a}$$
$$\dot{\underline{\eta}}_y = \underbrace{\left(\mathrm{D}F(s) + \mathrm{D}E(s) - \alpha\,\mathbf{1}\right)}_{\mathbf{P}}\underline{\eta}_y \quad . \tag{13b}$$

The stability of the primary system is not relevant for this analysis, as it is simply defined as a single trajectory of a chaotic system. However, the stability of the secondary, coupled system is of great interest, as it

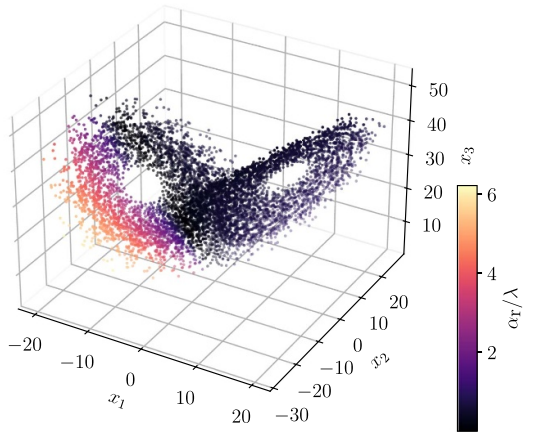


Figure 3. We show the minimal value of α_r, measured in Lyapunov exponents, required to ensure that all eigenvalues are negative for different choices of the synchronized state $\underline{s}$. (The synchronized states are, by definition, identical to the states of the primary system F.) The parameter setup is described in section 3.2.

is an indication for synchronization to the primary one. For this analysis, we perform the calculation of the matrix **P** in upper equation exemplary for the Lorenz63 system and arrive at

$$\mathbf{P} = \begin{pmatrix} -\sigma - \Delta_\sigma - \alpha & \sigma + \Delta_\sigma & 0 \\ \rho - \Delta_\rho - 2s_3 & 2 - \alpha & -2 \\ 2s_2 & 2s_1 & -\beta - \Delta_\beta - \alpha \end{pmatrix} . \tag{14}$$

Here, $(\sigma, \rho, \beta)^\intercal$ define the parameters of the true Lorenz63 system F, and $(\Delta_\sigma, \Delta_\rho, \Delta_\beta)^\intercal$ are the parameters of the Lorenz63 describing the error system E. $\underline{s}$ is an arbitrary synchronized state, meaning any state from F.

For F we use the standard parametrization of the Lorenz63 system. The parameters of E are chosen in such a way, that we arrive at our agnostic guess of $(1, 1, 1)^\intercal$ for the system G leading to $(\Delta_\sigma, \Delta_\rho, \Delta_\beta)^\intercal = (-9, -27, -5/3)$.

We can now calculate the eigenvalues of **P** for each synchronized state $\underline{s}$, which will be functions of the coupling strength α. The minimal value of the coupling strength for a synchronized state, for which all eigenvalues are negative, is denoted as α_r. If the coupling strength exceeds α_r, the perturbation around the synchronized state will decay exponentially fast.

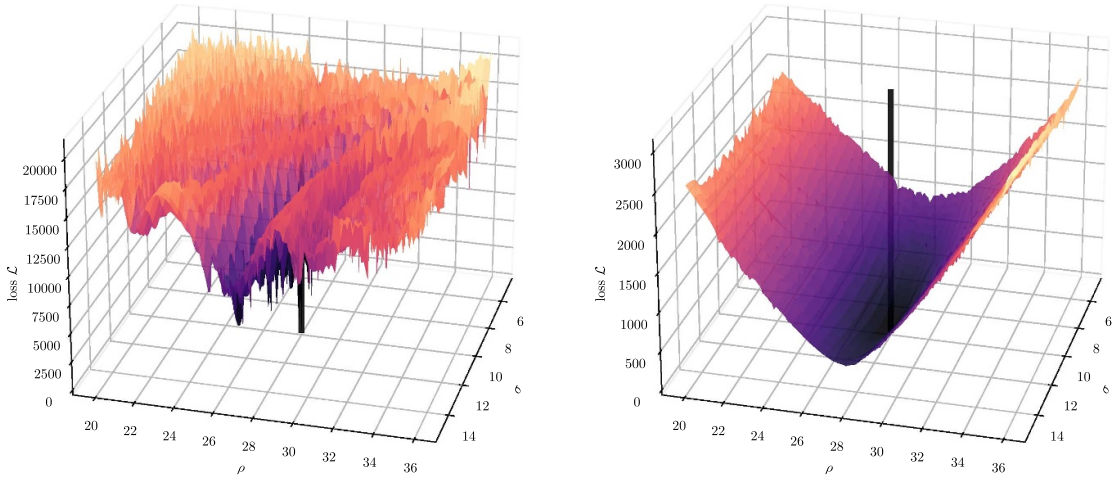
Figure 3 shows the calculation of α_r for all synchronized states of a Lorenz63 attractor described in previous setup. Here, we calculated the eigenvalues of **P** numerically as a function of α and solved for the first α, which yielded all negative eigenvalues: α_r. We note that the value α_r, which is required to ensure all negative eigenvalues across the whole attractor, does not exceed our typical choice of around ten Lyapunov exponents. This shows that the secondary system described by G does indeed synchronize to the primary, driving system, and this also reinforces our choice of a couple of Lyapunov exponents as coupling strength.

3.3. Loss function

We use a loss function $\mathcal{L}$ which describes the distance between the evolution of a primary system and the evolution of the secondary system as in Mariño and Míguez [28]. This means that the loss function is not dependent from the absolute states $\{\underline{x}\}$ and $\left\{\underline{y}\right\}$—instead, it is dependent from their derivatives.

The reason for this is as follows: in our algorithm we try to vary the parameters of a system in order to minimize a loss function. The states of each system are not explicitly dependent from the parameters. Rather, they depend on the integration method used. So by utilizing the derivative, we cut back on an intermediary step when applying the chain rule. Additionally by construction, we know the initial state of the secondary system, since it is the first data point available to us. If two systems have the same initial condition and the same derivative, they will evolve equally; being another argument for not requiring the absolute states of the system. In general, the loss function can be any norm in the mathematical sense. In practice, we find the mean-squared-error to be a useful error measure.

Instead of an online optimization as introduced by Mariño and Míguez [28], for our algorithm we use a fixed number of time steps to perform the optimization on. As we will show in sections 4.3 and 4.4, we find this change to improve the stability of the algorithm against uncertainties. This fixed number of steps is denoted 'evaluation length' l_e in this article.



a Uncoupled **b** Coupled

Figure 4. We show the loss $\mathcal{L}$ of two uncoupled Lorenz63 systems in (a), and the loss of two coupled Lorenz63 systems in (b). For the calculation of the loss we used an evaluation length of $l_e = 10^3$ and the systems were coupled using a coupling strength of $\alpha = 10$. The black line represents the parameters of the primary system. The figure in (b) shows a convex form on which a gradient descent-based optimizer can work on.

We calculate the loss as the sum of the mean-squared-error over each step in the evaluation length. The loss for each single time step i is ℓ_i, so we obtain the complete loss $\mathcal{L}$ by summing over all steps with

$$\ell_i = \left(\underline{\dot{x}}_i - \underline{\dot{y}}_i\right)^2 \tag{15a}$$

$$\mathcal{L} = \frac{1}{l_e}\sum_{i=1}^{l_e} \ell_i \quad . \tag{15b}$$

We note that the loss function has the secondary, coupled system in it, meaning the term with the coupling of equation (2*b*) is represented in $\underline{\dot{y}}$.

Experimentally, we noted that using the coupling we obtain a convex loss surface, as visualised in figure 4(b). There, we calculated the loss $\mathcal{L}$ for two uncoupled Lorenz63 systems (figure 4(a)) and two coupled Lorenz63 systems (figure 4(b)). For the primary system we used the standard parameters, while for the secondary system we set the parameter $\beta = \frac{8}{3}$ constant and swept through different choices for the parameters σ and ρ.

It is impossible to try to run an optimization for finding the minimum on the surface of the uncoupled systems in figure 4(a), since a clear direction towards the minimum of the loss, represented by the black line, cannot be found. However, the loss surface of the coupled systems (equations (2)) in figure 4(b) shows a convex shape, meaning that the application of gradient descent algorithms is justified and promising. We therefore can conclude that applying the coupling introduced in section 3.1 smoothens the loss surface making classical gradient descent algorithms applicable.

3.4. Update step

As we have shown in figure 4, utilizing the external coupling presented in section 3.1 transforms a non-convex loss surface to a convex one. Therefore, calculating the gradient of the loss function with respect to the parameters $\underline{\theta}$ and making steps in the negative direction of the gradient will lead to the set of parameters decreasing the loss function. Hence, with each step the primary system is described more precisely.

The loss function ℓ is not directly dependent from the parameters $\underline{\theta}$, instead the evolution of the secondary system $\underline{\dot{y}}$ is. Therefore, we need to apply the chain rule when calculating the derivative of the loss function with respect to a specific parameter θ in the parameter vector $\underline{\theta}$. Additionally, equation (15*b*) implies a sum over each coordinate n of an N-dimensional system, since we are dealing with vector-valued systems. So generally, the following equation for the j-th entry g_j of the gradient $\underline{g}$ holds:

$$g_j = \frac{\partial \mathcal{L}}{\partial \theta_j} = \frac{1}{l_e}\sum_{i=1}^{l_e} \frac{\partial \ell_i}{\partial \theta_j} \tag{16a}$$

$$\frac{\partial \ell}{\partial \theta_j} = -2 \sum_{n=1}^{N} (\dot{x}_n - \dot{y}_n) \frac{\partial \dot{y}_n}{\partial \theta_j} \quad . \tag{16b}$$

For the sake of readability we omitted the index i in equation (16b). It is understood however, that equation (16b) shows the loss term of a single sample i. Note also, that if a particular dimension n is not dependent on the parameter θ_j, that term will be 0 in the sum over the coordinates.

Performing the calculation for the Lorenz63 system leads to the following expression for the gradients of its parameters $\underline{\theta} = (\sigma, \rho, \beta)^\intercal$:

$$\frac{\partial \ell}{\partial \sigma} = -2\,(\dot{x}_1 - \dot{y}_1)\,(y_2 - y_1) \tag{17a}$$

$$\frac{\partial \ell}{\partial \rho} = -2\,(\dot{x}_2 - \dot{y}_2)\,y_1 \tag{17b}$$

$$\frac{\partial \ell}{\partial \beta} = +2\,(\dot{x}_3 - \dot{y}_3)\,y_3 \quad . \tag{17c}$$

Once we calculated the gradient, we need to update the system parameters in the direction minimizing the loss function. The most basic approach is utilizing the 'gradient descent' algorithm, where we subtract the direction with the negative slope from the parameters as done by Mariño and Míguez [28]:

$$\underline{\theta}_k = \underline{\theta}_{k-1} - \underline{\eta} \odot \underline{g}_k \quad . \tag{18}$$

Here, $\odot$ symbolises the Hadamard product between two vectors. The index indicates the k-th step of the optimization, and we perform the optimization until the all the parameters have converged. The vector $\underline{\eta}$ describes the learning rate, meaning the size of the update step we take after each optimization step k. Since chaotic systems can be more sensitive on certain parameters than on others, we choose a different learning rate for each parameter. Heuristically, we discovered that setting the learning rate for each parameter in such a way, that the change of the first update step is of order 10^{-1}, leads to a good performance.

3.5. Modern optimizers

In this section we will briefly introduce two all-purpose optimizers and analyze their performance against the plain gradient descent method, since the rise of machine learning led to the development of more sophisticated first-order optimizers.

A modern optimizer 'Adam' introduced by Kingma and Ba [29] utilizes an estimation of the first and second moment of the gradient in its calculation. Those estimates are obtained by exponentially weighting past gradients (first moment) and past squared gradients (second moment). The first moment is captured in the vector $\underline{m}$ and the second moment is captured in the vector $\underline{v}$. With $\underline{g}_k$ being the gradient at optimization step k, the moment estimations are updated as described in [29] using

$$\begin{aligned} \underline{m}_k &= \beta_1\, \underline{m}_{k-1} + (1-\beta_1)\, \underline{g}_k \\ \underline{v}_k &= \beta_2\, \underline{v}_{k-1} + (1-\beta_2)\, \left(\underline{g}_k \odot \underline{g}_k\right) \quad . \end{aligned}$$

Both vectors, $\underline{m}$ and $\underline{v}$, are initialized with $\underline{0}$. Kingma and Ba [29] noted that by doing so, the estimates are biased towards $\underline{0}$. For that reason, they propose the following bias-corrected estimate:

$$\begin{aligned} \hat{\underline{m}}_k &= \frac{1}{1-\beta_1^k}\, \underline{m}_k \\ \hat{\underline{v}}_k &= \frac{1}{1-\beta_2^k}\, \underline{v}_k \quad . \end{aligned}$$

The superscript k actually represents taking the value to the k-th power. This results in the update step

$$\underline{\theta}_k = \underline{\theta}_{k-1} - \underline{\eta} \odot \hat{\underline{m}}_k \oslash \left(\sqrt{\hat{\underline{v}}_k} + \varepsilon\right) \quad . \tag{19}$$

Here, $\oslash$ symbolises the element-wise division and the square-root is understood to be taken over each element of $\hat{\underline{v}}_k$. The step size for each parameter $\underline{\eta}$ has been estimated using the method discussed in section 3.4. We adopt the recommendations by Kingma and Ba [29] for the remaining free parameters: $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\varepsilon = 10^{-8}$. As can be seen in figure 5, this optimization works and leads to convergence of the parameters.

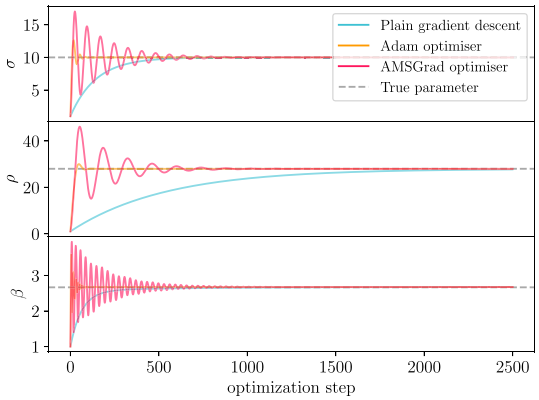


Figure 5. Here, we show the evolution of the coefficients for different optimizers applied on the Lorenz63 system, and we can see the modern optimizers (equations (19) and (20)) outperforming the vanilla gradient descent (equation (18)). We use a Lorenz63 system with an initial state of $(5,5,5)^\intercal$ and a time step of $\Delta t = 10^{-3}$ as primary system. For an equal comparison, the hyperparameters for each optimization algorithm were kept constant and had following values: $l_e = 10^3$, $\alpha = 10^3$. We used an agnostic initial guess of 1 for each parameter to be estimated in the secondary system.

Another optimizer we want to introduce in this article is the 'AMSGrad' optimizer by Reddi *et al* [30]. In their article the authors found scenarios in which the Adam optimizer does not converge to the optimal solution [30]. They show that this is based on considering the moving average of past squared gradients—one of the key features of Adam. The issue is, that under certain circumstances the Adam optimizer may take steps too large [30]. This happens if the second moment estimate $\hat{v}_k^j$ for a parameter j at step k gets too small. They mitigated this issue by considering the element-wise maximum value of the new estimate and the previous estimate. This way they ensure that the step size will not increase for each parameter:

$$\hat{v}_k^j = \max\left(\hat{v}_{k-1}^j, v_k^j\right) \quad \forall j \in \{1, \ldots, \dim \underline{\theta}\} \quad .$$

Additionally, they simplify the optimizer by discarding the bias correction terms. Therefore, the complete update step can be calculated using

$$\begin{aligned} \underline{m}_k &= \beta_1 \underline{m}_{k-1} + (1-\beta_1)\, \underline{g}_k \\ \underline{v}_k &= \beta_2 \underline{v}_{k-1} + (1-\beta_2)\left(\underline{g}_k \odot \underline{g}_k\right) \\ \hat{\underline{v}}_k &= \max\left(\hat{\underline{v}}_{k-1}, \underline{v}_k\right) \\ \underline{\theta}_k &= \underline{\theta}_{k-1} - \underline{\eta} \odot \underline{m}_k \oslash \sqrt{\hat{\underline{v}}_k}. \end{aligned} \tag{20}$$

The maximum function in upper equation is applied element-wise for each variance estimation of each parameter. Again, we use the recommended values for the remaining coefficients [30]: $\beta_1 = 0.99$, $\beta_2 = 0.999$. Unlike the original article, we use a different reference learning rate $\underline{\eta}$ for each parameter as derived in section 3.4.

We compared the performance for each optimizer and show the result in figure 5. There we can see that all optimization algorithms eventually converge to the right solution. However, the sophisticated optimizers converge faster than plain gradient descent. Additionally, we note that the AMSGrad optimizer oscillates around the true parameters while the Adam optimizer does not show any. In our studies we did not note any negative effect on the optimization from the oscillations. In the following we will use the AMSGrad optimizer, since it converges in more cases than the Adam optimizer [30].

4. Results

We applied our new algorithm on an ample selection of different, synthetic, chaotic systems and found promising results in each of them. In this section we will present our findings for those synthetic systems and explore the limitations of our algorithm.

For assessing the quality of our optimization, we monitor two metrics: firstly, we calculate the predictable time t_p. It describes the time after which the prediction differs substantially from the true time series. Since the absolute predictable time is highly correlated to the largest Lyapunov exponent λ of the system, we

Table 1. This table shows the metrics we monitor for different, three-dimensional systems. For each system we repeat the optimization five times and report the mean and one standard deviation. For each run we used an integration step of $\Delta t = 10^{-3}$, a coupling strength of $\alpha = 10^3$, and an evaluation length of $l_e = 10^4$.

System	Parameters	Parameter error E_θ	Predictable time τ_p
Thomas [36]	$f(x_i) = \sin x_i$, $b = 0.21$	$1.3 \pm 3 \times 10^{-5}$	3.1 ± 0.2
Sprott [37]	$a = 2.07$, $b = 1.79$	$3.1 \pm 1.2 \times 10^{-4}$	9.8 ± 0.8
Lorenz63 [2]	$\sigma = 10$, $\rho = 28$, $\beta = \frac{8}{3}$	$4.2 \pm 0.5 \times 10^{-3}$	6.88 ± 0.05
Dadras–Momeni [38]	$a = 3$, $b = 2.7$, $c = 1.7$, $d = 2$, $e = 9$	$7.9 \pm 1.7 \times 10^{-3}$	1.89 ± 0.05
Rössler [39]	$a = 0.1$, $b = 0.1$, $c = 14$	0.011 ± 0.015	4.9 ± 0.2
Halvorsen [40]	$a = 1.89$	0.0147 ± 0.0007	1.094 ± 0.002
Lorenz86 [41]	$a = 0.25$, $b = 4$, $f = 1.1$, $g = 8$	0.023 ± 0.006	0.75 ± 0.05
Three-scroll unified [42]	$a = 40$, $c = \frac{5}{6}$, $d = 0.5$, $e = 0.65$, $f = 20$	0.037 ± 0.002	0.3 ± 0.2

measure it in units of the Lyapunov time τ_λ for ensuring comparability among different systems using

$$\tau_p = \frac{t_p}{\tau_\lambda} = t_p \lambda \quad .$$

For our work, we define a prediction as wrong if it differs substantially from the true time series and follows a different path on the attractor. Small deviations of the prediction from the true time series are acceptable as long as they do not persist. For this reason, we define the predictable time, as that time when the error in the interval $[t_p,\ t_p + \frac{1}{10}\tau_\lambda]$ is for the first time bigger than 15% of the scale s of the system, leading to

$$s = \frac{1}{\tau_\lambda} \int_0^{\tau_\lambda} \|\underline{x}(t)\| \, \mathrm{d}t$$
$$\int_{t_p}^{t_p + \frac{1}{10}\tau_\lambda} \|\underline{x}(t) - \underline{\hat{x}}(t)\| \, \mathrm{d}t > 15\% s \quad .$$

Here, $\underline{x}$ describes the true time series and $\underline{\hat{x}}$ describes the predicted time series. By using this metric, we can compare different systems with each other as this metric relies solely on system properties and uses the characteristic time and size of each system. The fractions in this were chosen heuristically to reflect the requirement to allow small temporary deviations, but trigger at bigger deviations over a period of time.

The second metric we monitor is the average mean-absolute-error E_θ of the fitted parameters $\underline{\hat{\theta}}$ with respect to the real parameters $\underline{\theta}$ described by

$$E_\theta = \frac{1}{\dim \underline{\theta}} \sum_{i=1}^{\dim \underline{\theta}} \left| \theta_i - \hat{\theta}_i \right| \quad .$$

This metric allows us to understand the numerical precision of our optimization.

The results for different, three-dimensional, chaotic systems are shown in table 1. We see that the parameter reconstruction is precise to the first digit and using this precision we are able to predict the systems a handful of Lyapunov times ahead. Additionally, we note that our optimization algorithm is stable against the initial condition and hence the position on the attractor. The results are also stable against the exact choice for the hyperparameters α and l_e, as long as they are in sensible ranges.

4.1. High-dimensional system

So far, we only applied our optimization algorithm on three-dimensional systems with a handful of parameters. In the following we want study the applicability of this optimization on high-dimensional problems. For this purpose, we consult the Lorenz96 system [43], as it describes a system where the dimensionality can be increased as desired. All the coefficients in front of the variables have the equal value 1 and an N-dimensional system is defined using

$$\dot{x}_d = (x_{d+1} - x_{d-2})\, x_{d-1} - x_d + F \quad d \in \{1, \ldots, N\} \quad .$$

We assume cyclic boundary conditions described by $x_{-1} = x_{N-1}$, $x_0 = x_N$, and $x_{N+1} = x_1$.

We need to extend this model for our purposes, since our initial guess for the coefficients almost perfectly aligns with the real parameters. For this, we introduce a parameter in front of each variable in each

Table 2. This table shows the result for the Lorenz96 system with an increasing number of dimensions. We observed some runs in which the predictable time contained outlier to the positive direction. In order to not skew our results to the better, we reported the minimal predictable time for each set of dimensionality.

Dimension N	Number parameters	E_θ	$\min \tau_\mathrm{p}$
5	16	0.07 ± 0.07	2.3
6	19	$2.4 \pm 1.2 \times 10^{-3}$	6.3
7	22	$1.3 \pm 1.2 \times 10^{-3}$	8.1
8	25	$7 \pm 8 \times 10^{-3}$	8.0
9	28	$2.9 \pm 1.9 \times 10^{-3}$	3.6
10	31	0.05 ± 0.07	3.2
12	37	$4 \pm 3 \times 10^{-3}$	2.0
14	43	0.04 ± 0.05	2.0
16	49	$7 \pm 6 \times 10^{-3}$	1.5
32	97	0.05 ± 0.04	2.1
64[a]	193	1.04×10^{-3}	2.0

[a] Due to the high computational resources required, only one experiment was performed.

dimension, which leads to the following, parametrized Lorenz96 system:

$$\dot{x}_d = \left(p_{3(d-1)+1}\, x_{d+1} - p_{3(d-1)+2}\, x_{d-2}\right) x_{d-1} - p_{3(d-1)+3}\, x_d + p_{3N+1} \quad d \in \{1, \ldots, N\} \quad .$$

The parameters are stored in a $(3N+1)$-dimensional vector $\underline{p}$. Note that in our parametrized Lorenz 96 system we followed the original Lorenz96 system and kept the forcing constant F equal among each dimension d. We stored the forcing constant as the last value in the vector $\underline{p}$. The parameters in front of the variables, p_1 to p_{3N}, were drawn from a uniform distribution in interval $[0, 1]$ and the forcing constant, p_{3N+1}, was assigned the value 8 in accordance to the original model [43].

We performed our optimization for different Lorenz96 systems with an increasing dimensionality. For each dimensionality we performed five experiments and used a different parameter vector $\underline{p}$ each time. We used a fine time step of $\Delta t = 10^{-4}$, an evaluation length $l_\mathrm{e} = 10^5$, and a coupling strength of $\alpha = 10^5$. The result of the optimizations are shown in table 2. There we can see that the optimization works for a wide range of dimensions and is reproducible for vectors with different random parameters. We show that our optimization algorithm is also able to fit parameters with their count being in the order of hundreds. Table 2 also shows that the predictability does not suffer with an increasing dimensionality. For each dimension we are able to predict a couple of Lyapunov times ahead, as we are able with the three-dimensional systems in table 1. Instead, we notice that our algorithm does not prefer a specific region of dimensions. This demonstrates the applicability of this algorithm to high-dimensional systems with a large number of free parameters.

4.2. Robustness against sampling size

For our theoretical derivation we assumed an evenly spaced time grid. However, in this section we want to analyze the robustness of our algorithm against the sampling size. We want to answer two questions: firstly, we study whether the size of the time grid matters, and secondly, we study the robustness of our algorithm against an uneven time grid. Latter question is especially interesting for potential real world applications, as an even time grid cannot be necessarily assumed in measured data.

In order to study different sampling sizes, we perform our optimization for different step sizes Δt and report the performance, the mean absolute parameter error E_θ. Here, we distinguish two scenarios. In the first scenario we use an evaluation length of $l_\mathrm{e} = 1\,000$ for all sampling sizes Δt. For the second scenario we used an equal amount of passed time, 10, for each Δt, resulting in an evaluation length of $l_\mathrm{e} = \frac{10}{\Delta t}$. The results are shown in figure 6, where we see that our algorithm performs well for all commonly used sampling sizes. Consistent to our remaining results, we find no strong dependence on the training length.

For studying the effects of an uneven grid we simulate the Lorenz63 system using a base step size of Δt. We iterate through each data point and with a probability of p we skip the next data point. The time step sizes present in the data now follow a geometric distribution with their mean value given by $\frac{\Delta t}{1-p}$ and their standard deviation by $\frac{\Delta t \sqrt{p}}{1-p}$. We collect data points until we reach an evaluation length of $l_\mathrm{e} = 1\,000$. The results are shown in figure 7, where we performed a hundred realizations for each probability p and represent each data point by its mean sampling size. There we see that we can expect a reasonable estimate of the parameters, as long as the mean time step size does not differ more than a few percent from the actual time step size. In our experiments we found no difference in results whether we simulate the secondary system with the mean step size or the actual step size. However, interestingly we found a worse performance for a

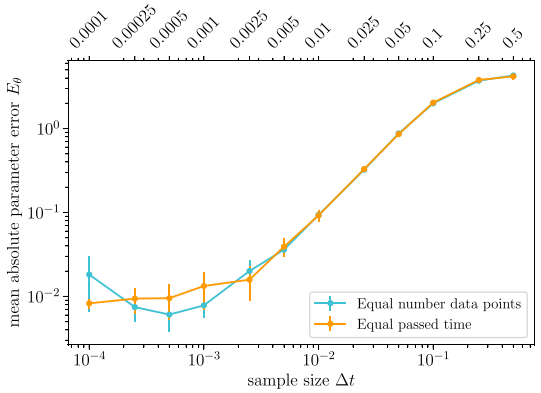


Figure 6. We show the performance on the Lorenz63 system for different sampling sizes and report the average over ten realizations. The blue line uses an evaluation length of $l_e = 1\,000$ and the orange one an evaluation length of $l_e = \frac{10}{\Delta t}$. Each experiment is done with a coupling strength of $\alpha = 10^2$.

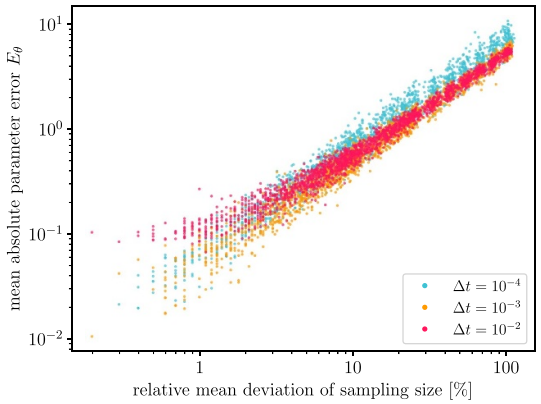


Figure 7. We show the performance on the Lorenz63 system for different mean sampling sizes relative to their original sampling size. For this plot we used skipping probabilities p ranging from 1 % to 50 % and a coupling strength of $\alpha = 10^2$.

very small grid size. While an uneven time grid with small disturbances can lead to a reasonable estimate of the parameters, we require an even grid for a precise reconstruction of the parameters.

4.3. Robustness against noise

So far, all calculations have been performed on synthetic systems, which are precise and error-free by construction. In this section, we analyze the robustness of our optimization against noise in the data making up the primary system. For that, we simulate the classical Lorenz63 system with its standard parameters and add a Gaussian noise on each data point and each coordinate. In our experiments, the standard deviations σ for the noise range from 0.01 to 2. Before adding the Gaussian noise to the precise data, we calculate the signal-to-noise ratio (SNR) as the ratio of the power of the signal to the power of the noise [44]. With the power being calculated as the squared amplitude A, we can express the SNR using

$$\mathrm{SNR} = \left(\frac{A_{\mathrm{signal}}}{A_{\mathrm{noise}}}\right)^2$$

$$\mathrm{SNR} = 10\log_{10}\left(\frac{A_{\mathrm{signal}}}{A_{\mathrm{noise}}}\right)^2 \mathrm{dB} \quad .$$

The amplitude A of a time series $\underline{x}(t)$ can be calculated with

$$A = \sqrt{\frac{1}{t}\int_0^t \|\underline{x}(\tilde{t})\|^2 \,\mathrm{d}\tilde{t}} \quad .$$

We ran two distinct experiments and in the first we analyzed the dependency of the noisy data on the coupling strength, and the other time the dependency on the evaluation length. We use the Lorenz63 system for the following analyses.

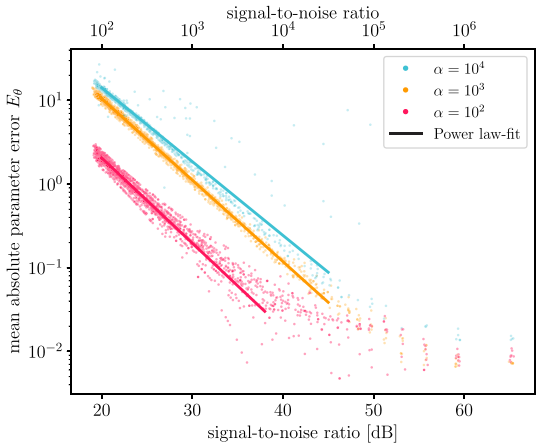


Figure 8. We show the performance for different coupling strengths on different levels of noise. We use an evaluation length of $l_e = 10^4$. On noisy data lower coupling strengths work better.

4.3.1. Dependency on hyperparameters

Our algorithm has two hyperparameters: the coupling strength and the evaluation length. The results for varying the coupling strength are shown in figure 8. An interesting pattern emerges: the parameter errors E_θ seem to follow a power law in a certain region. The cut-off of this power law-behaviour at an error of about 10^{-2} stems from our convergence criterion for these runs. We observe that a lower coupling strength leads to a more negative exponent in the fit of said, observed power law. This implies a better performance for lower coupling strengths, since it means that the error does not grow as quickly with a rising noise level. In practice when dealing with real data, this means that setting the coupling strength too high will yield worse results than a lower coupling strength. One possible explanation could be that a strong coupling forces the secondary system to treat the noise of the primary system as actual dynamics, whereas a less strong coupling allows the dynamics term $F\left(\underline{y}\right)$ in equation (2*b*) to be more dominant.

4.3.2. Comparison against benchmark

Due to their similarity, we use the algorithm by Mariño and Míguez [28] as a benchmark against our algorithm. Additionally, we compare our algorithm against the SINDy algorithm by Brunton *et al* [11]. Basically, the SINDy algorithm uses the numerical derivative of a measured time series and regresses it sparsely against a library of linear and nonlinear terms. We refer to [11] for a detailed explanation. For our experiments, we build the library to only consist of the respective terms in the Lorenz63 equations. Since their algorithm utilizes the same prior knowledge while being conceptionally simpler, we believe it to be a fair benchmark.

We parameterize our algorithm with a moderate coupling strength of $\alpha = 10^2$, based on our findings from previous subsection. For our algorithm and SINDy we use 1 000 data points and for the algorithm by Mariño and Míguez [28] we use as many steps as it needs to converge.

We show our results in figure 9: while the algorithm by Mariño and Míguez [28] occasionally outperforms our algorithm, we find our algorithm to be more stable and producing better results more reliably—especially for higher levels of noise. Comparing our results to SINDy, we find our algorithm being better suited for noisy data, while SINDy performs better on very low levels of noise. As in previous section 4.3.1, we find our algorithm to flatten out at an error of 10^{-2} due to our convergence criterion. While setting a stricter convergence criterion would lead to comparable results to SINDy, it would take orders of magnitude more time.

4.4. Robustness against incorrect equations

A second source of uncertainty may stem from an incorrect knowledge of the governing equations. In order to test this, we introduce an additional linear and a nonlinear term to the classical Lorenz63 system resulting in the system described by

$$\dot{x}_1 = -\sigma x_1 + \sigma x_2 + \kappa x_2 x_3 \tag{21a}$$

$$\dot{x}_2 = \rho x_1 - x_2 - x_1 x_3 + \xi x_2 \tag{21b}$$

$$\dot{x}_3 = -\beta x_3 + x_1 x_2 \quad . \tag{21c}$$

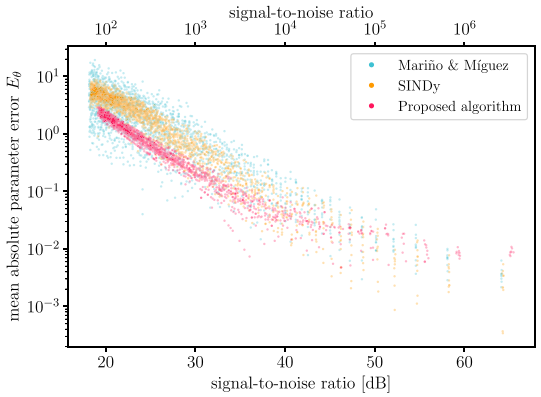


Figure 9. We compare the performance of our algorithm (red) against the algorithm by Mariño and Míguez [28] (blue) and SINDy [11] (orange).

Table 3. This table shows the result for performing our proposed optimization on data stemming from the Lorenz63 system onto the system described by equations (21). We performed the experiment fifty times with differing initial conditions and the error shows one standard deviation of the values.

Parameter	True Value	Reconstructed Value
σ	10	10.027 ± 0.015
ρ	28	27.795 ± 0.014
β	$2.\bar{6}$	2.676 ± 0.003
κ	0	-0.0014 ± 0.0004
ξ	0	0.000 ± 0.002

This adds the following two gradients to the set of equations in equations (17):

$$\frac{\partial \ell}{\partial \kappa} = -2\,(\dot{x}_1 - \dot{y}_1)\,y_2\,y_3$$
$$\frac{\partial \ell}{\partial \xi} = +2\,(\dot{x}_2 - \dot{y}_2)\,y_3 \quad .$$

As in the other experiments, we initialized all parameters with one, since we have no prior knowledge of the parameters. Using a coupling strength of $\alpha = 10^2$ and an evaluation length of $l_e = 10^3$, we find the parameters as shown in table 3. We see that our algorithm is able to correctly set the parameters to almost zero and minimize their effect on the governing equations. The remaining, true parameters of the system are reconstructed correctly as well.

Applying this problem setup to the algorithm by Mariño and Míguez [28] leads to a divergence of the coefficients. On the other hand, SINDy can deal easily with additional wrong terms and correctly sets them to a value close to zero.

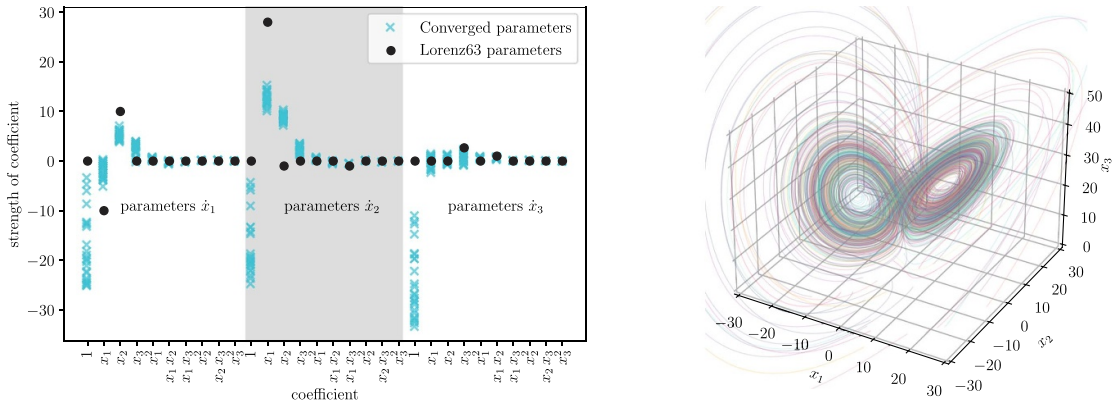
At this point we want to mention two limitations of our algorithm: The coefficients κ and ξ are set correctly to a value close to zero. However, so far we do not have a possibility to determine whether the terms with a small coefficient can be excluded or are relevant for the dynamics of the system. Therefore, we advise against discarding terms, which have a coefficient close to zero.

Additionally, our algorithm is susceptible to an under-inclusion of terms. This means that if a term is not present in the prior knowledge of the system, our algorithm will not work and will diverge instead. However, this can be resolved easily as the reconstruction algorithm by Ma *et al* [9] tends to over-include terms, rendering this a non-issue.

4.4.1. Application to a skeleton system

This method has been developed for finding the parameters of governing equations, given the terms were already known. So far, all analyses have been performed under this framework. In previous section, we have seen that our algorithm is stable against the inclusion of two wrong terms.

However, for this section we change the framework and assume no prior knowledge on the terms of the governing equations. This is an extreme example of incorrect prior knowledge. We simply provide a library of possible terms the system is allowed to have. The library is defined by all coordinate permutations up to a



a Parameters for second order library

b Trajectories

Figure 10. We show the results for the optimization of fitting the Lorenz63 data on the second order skeleton system. We performed 24 runs out of which each parameters result in the butterfly shape starting from random initial conditions. Figure (a) shows the strength of the coefficient for each dimension. Figure (b) shows the integrated trajectories. The initial condition is a random number between −20 and 20 for each coordinate. For each set of parameters two trajectories are shown differing in their initial condition.

certain order. For a library of second order, the prior knowledge is represented in equations (22), where we store the coefficients in the matrix $\boldsymbol{\Theta}$. For the optimization we use the vector representation.

$$\dot{x}_1 = \theta_{11} + \theta_{12}x_1 + \theta_{13}x_2 + \theta_{14}x_3 + \theta_{15}x_1^2 + \theta_{16}x_1x_2 + \theta_{17}x_1x_3 + \theta_{18}x_2^2 + \theta_{19}x_2x_3 + \theta_{110}x_3^2 \tag{22a}$$

$$\dot{x}_2 = \theta_{21} + \theta_{22}x_1 + \theta_{23}x_2 + \theta_{24}x_3 + \theta_{25}x_1^2 + \theta_{26}x_1x_2 + \theta_{27}x_1x_3 + \theta_{28}x_2^2 + \theta_{29}x_2x_3 + \theta_{210}x_3^2 \tag{22b}$$

$$\dot{x}_3 = \theta_{31} + \theta_{32}x_1 + \theta_{33}x_2 + \theta_{34}x_3 + \theta_{35}x_1^2 + \theta_{36}x_1x_2 + \theta_{37}x_1x_3 + \theta_{38}x_2^2 + \theta_{39}x_2x_3 + \theta_{310}x_3^2 \tag{22c}$$

We perform the optimization on the Lorenz63 system with the standard parametrization using a step size of $\Delta t = 10^{-3}$, a coupling strength of $\alpha = 10^3$, and an evaluation length of $l_e = 10^4$. In each run we differ the starting point of the Lorenz63 system, to get variability in the training data. For our experiments we use different libraries starting from a library containing only first order terms up to a library containing terms up to the fifth order. The results are presented in the following.

First order library

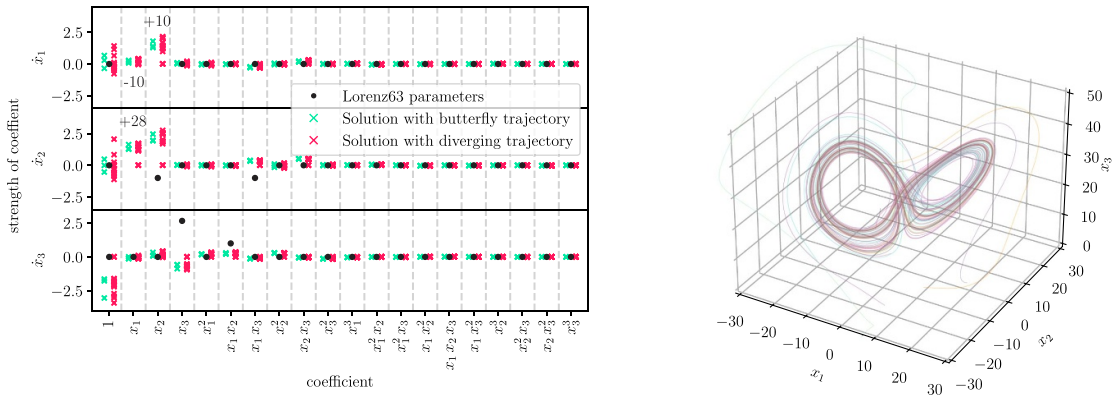
For a library consisting only of linear terms, the loss function of our proposed algorithm still converges. However, the integrated solutions do not describe the Lorenz63 system in any way. Even in the in the training data the integrated solution differs wildly from the Lorenz63 input. Linear equations are not capable to reproduce the nonlinear dynamics of the Lorenz63 system. This serves as a cautionary note for our algorithm: the loss converging does not necessarily imply that the reached solution is correct. We note that the converged parameters are not unique, and it seems that multiple local minima exist using a first order library.

Second order library

For a library consisting of terms up to second our algorithm converges for every of our 24 different initial conditions. However, the found parameters are not unique and also do not match the input parameters of the classical Lorenz63 system.

Instead, we find a huge spread in the coefficients of constant and first order terms, as can be seen in figure 10(a). In figure 10(b) we show that the integrated solutions still result in the typical Lorenz63 attractor. Like the classical parametrization, our found parameters have their largest Lyapunov exponent greater 0 laying between $\lambda_{\mathcal{O}(2)} \in [0.60, 0.78]$; and the correlation dimension $C_{\mathcal{O}(2)} \in [1.91, 2.09]$ lies in the range of the classical Lorenz63 system's. This implies that the Lorenz63 shape does not strongly depend on zeroth and first order terms, and it is the nonlinear terms, which are important to define the butterfly shape. For our algorithm it means that multiple local minima exist, when the library is chosen too coarse.

This raises an important question about the uniqueness of the Lorenz63 equations. We find this question to be of special importance, especially for black box-machine learning models: which set of equations are they really learning—the real parameters or a set of parameters also producing *an* attractor?



a Parameters for third order library

b Trajectories for converging set of parameters

Figure 11. We show the results for the optimization of fitting the Lorenz63 data on the third order skeleton system. We performed 18 runs out of which four sets of parameters result in the butterfly shape starting from random initial conditions. The remaining ones diverge, even if only for some starting conditions. Figure a shows the strength of the coefficient for each dimension. Figure b shows the integrated trajectories. The initial condition is a random number between −20 and 20 for each coordinate. For each set of parameters five trajectories are shown differing in their initial condition.

Third order library

Integrating the solutions of the third order library leads to either diverging paths or paths producing a butterfly-shaped attractor. In figure 11(a) we show the converged parameters. There we note, that we are unable to find a discriminating set of rules for the parameters determining whether their trajectories diverge or not. However, if the trajectory does not diverge, we reach the butterfly-shaped attractor shown in figure 11(b). The attractors have Lyapunov exponents of $\lambda_{\mathcal{O}(3)} \in [0.04, 0.35]$ and correlation dimensions of $C_{\mathcal{O}(3)} \in [1.60, 1.61]$. This underlines our finding that multiple sets of equations are able to produce the Lorenz63 shape.

Fourth and fifth order library

For libraries containing terms up to the fourth and fifth order respectively, we find the training process to be unstable. As before, the converged parameters are not unique indicating multiple local minima. While some converged parameters provide a good local description of the Lorenz63 system, none of the parameters are able to capture the long-term dynamic of the original equations. Comparing the results to the high-dimensional Lorenz96 system shown in table 2, we conclude that the problem is not the high number of free parameters, but instead the incorrect knowledge of the structure of the governing equations.

5. Discussion

The aim of this work was to develop an algorithm for estimating the parameters of chaotic systems using data. The posed problem was rather specific and asked whether it was possible to reconstruct the parameters, if the structure of the governing equations and a time series stemming from the target system were known. In this work we have shown that it is possible to reconstruct the parameters given those assumptions, and we presented a framework for applying this on arbitrary systems. For achieving this, we borrowed ideas from synchronization and machine learning. Specifically, we coupled a secondary system, of which we can control the parameters, to a primary system being entirely described by the data. The secondary system encodes our assumptions on the structure of governing equations. We then modified the parameters of the secondary system using gradient descent-based algorithms to minimize the loss between the two systems. The values of the parameters minimizing the loss are a good estimate of the parameters describing the real data.

In this work we have shown that our algorithm is applicable to a plethora of different synthetic systems with prediction performances of a couple Lyapunov times. Additionally, we have shown that this algorithm also works in a high-dimensional space with a large number of free parameters. Up to a certain extent, our algorithm is robust against noise in the data and incorrect assumptions of the structure of the governing equations. We are more stable than the comparable algorithms.

While we have demonstrated the applicability of our algorithm to synthetic data with controlled defects, studies and applications to real-world data still remain to be seen. A big challenge with real-world data is the number of observations. While our algorithm requires a number of data points in the order of 10^3, publicly available real-world data sets stemming from chaotic models typically do not have these number of

observations. Applying our algorithm to the classic *Paramecium*/*Didinium* data set [45] remains unsuccessful due to the ultra-low number of 71 observations.

Our algorithm is not capable of reconstructing the exact governing equations given only a skeleton of terms up to a certain order. However, while the solutions are not correct regarding the found parameters, the found solutions still behave like the ground truth data. This opens the question to the uniqueness of governing equations of attractors. Especially with respect to black box-machine learning models: which representation of the attractor are they actually learning? Additionally, further work is required to address the question whether this optimization also works with weaker types of synchronization, like phase synchronization [46].

This work can be embedded in the greater topic of reconstructing governing equations from time series. Using Ma *et al* [9] for deriving the structure of the governing equations and afterwards this algorithm for calibrating the coefficients can be a way to infer the dynamical equations describing the data at hand. This method could then be used across different disciplines for making interpretable, data-driven forecasts.

Data availability statement

All data that support the findings of this study are included within the article (and any supplementary files).

Acknowledgments

We would like to thank Allianz Global Investors for providing computational resources.

Conflict of interest

The authors have no conflicts to disclose.

ORCID iDs

Davide Prosperino https://orcid.org/0009-0005-3878-2460
Haochun Ma https://orcid.org/0009-0008-5894-0448
Christoph Räth https://orcid.org/0000-0002-5545-3029

References

[1] Lukoševičius M and Jäger H 2009 Reservoir computing approaches to recurrent neural network training *Comput. Sci. Rev.* **3** 127–49
[2] Lorenz E N 1963 Deterministic nonperiodic flow *J. Atmos. Sci.* **20** 130–41
[3] Mihailović D T, Mimić G and Arsenić I 2014 Climate predictions: the chaos and complexity in climate models *Adv. Meteorol.* **2014** 878249
[4] Ruelle D and Takens F 1971 On the nature of turbulence *Commun. Math. Phys.* **20** 167–92
[5] Engbert R and Drepper F R 1994 Chance and chaos in population biology–models of recurrent epidemics and food chain dynamics *Chaos Solitons Fractals* **4** 1147–69
[6] Billings L and Schwartz I B 2002 Exciting chaos with noise: unexpected dynamics in epidemic outbreaks *J. Math. Biol.* **44** 31–48
[7] Mandelbrot B 1963 The variation of certain speculative prices *J. Bus.* **36** 394–419
[8] Peters E E 1994 *Fractal Market Analysis: Applying Chaos Theory to Investment and Economics* (Wiley)
[9] Ma H, Haluszczynski A, Prosperino D and Räth C 2022 Identifying causality drivers and deriving governing equations of nonlinear complex systems *Chaos* **32** 103128
[10] Daniels B C and Nemenman I 2015 Automated adaptive inference of phenomenological dynamical models *Nat. Commun.* **6** 8133
[11] Brunton S L, Proctor J L and Kutz J N 2016 Discovering governing equations from data by sparse identification of nonlinear dynamical systems *Proc. Natl Acad. Sci.* **113** 3932–7
[12] Champion K, Lusch B, Kutz J N and Brunton S L 2019 Data-driven discovery of coordinates and governing equations *Proc. Natl Acad. Sci.* **116** 22445–51
[13] Tao C, Zhang Y and Jiang J J 2010 Estimating system parameters from chaotic time series with synchronization optimized by a genetic algorithm *Phys. Rev. E* **76** 016209
[14] Mukhopadhyay S and Banerjee S 2012 Global optimization of an optical chaotic system by chaotic multi swarm particle swarm optimization *Expert Syst. Appl.* **19** 917–24
[15] Jafari S, Sprott J C, Pham V-T, Golpayegani S M R H and Jafari A H 2014 A new cost function for parameter estimation of chaotic systems using return maps as fingerprints *Int. J. Bifurcation Chaos* **34** 1450134
[16] Bagnoli F and Baia M 2023 Synchronization, control and data assimilation of the Lorenz system *Algorithms* **16** 213
[17] Fujisaka H and Yamada T 1985 Stability theory of synchronized motion in coupled-oscillator systems *Prog. Theor. Phys.* **69** 32–47
[18] Pecora L M and Carroll T L 1990 Synchronization in chaotic systems *Phys. Rev. Lett.* **64** 821–4
[19] Parlitz U 1996 Estimating model parameters from time series by autosynchronization *Phys. Rev. Lett.* **76** 1232–5
[20] Parlitz U, Junge L and Kocarev L 1996 Synchronization-based parameter estimation from time series *Phys. Rev. E* **54** 6253–9
[21] Maybhate A and Amritkar R E 1999 Use of synchronization and adaptive control in parameter estimation from a time series *Phys. Rev. E* **59** 284–93
[22] Huang D and Guo R 2004 Identifying parameter by identical synchronization between different systems *Chaos* **14** 152–9

[23] Sun F, Peng H, Luo Q, Li L and Yang Y 2009 Parameter identification and projective synchronization between different chaotic systems *Chaos* **19** 023109
[24] Wang C, He Y, Ma J and Huang L 2014 Parameters estimation, mixed synchronization and antisynchronization in chaotic systems *Complexity* **20** 64–73
[25] Lü L and Wei Q 2019 Parameter estimation and synchronization in the uncertain financial network *Physica A* **535** 122418
[26] Abarbanel H D I, Creveling D R, Farsian R and Kostuk M 2009 Dynamical state and parameter estimation *SIAM J. Appl. Dyn. Syst.* **8** 1341–81
[27] Creveling D R, Jeanne J M and Abarbanel H D I 2008 Parameter estimation using balanced synchronization *Phys. Lett. A* **372** 2043–7
[28] Mariño I P and Míguez J 2006 An approximate gradient-descent method for joint parameter estimation and synchronisation of coupled chaotic systems *Phys. Lett. A* **351** 262–7
[29] Kingma D P and Ba J L 2015 Adam: a method for stochastic optimization *3rd Int. Conf. on Learning Representations, ICLR 2015* (*San Diego, CA, USA*, *7–9th May 2015*) ed Y Bengio and Y LeCun
[30] Reddi S J, Kale S and Kumar S 2019 On the convergence of Adam and beyond
[31] Hindmarsh A C and Petzold L R 2005 *LSODA, Ordinary Differential Equation Solver for Stiff or Non-Stiff System*
[32] Eroglu D, Lamb J S W and Pereira T 2017 Synchronisation of chaos and its applications *Contemp. Phys.* **58** 207–43
[33] Leonov G A 2008 *Strange Attractors and Classical Stability Theory* (St. Petersburg University Press)
[34] Skokos C 2010 The Lyapunov characteristic exponents and their computation *Dynamics of Small Solar System Bodies and Exoplanets* ed J J Souchay and R Dvorak (Springer)
[35] Pecora L M and Carroll T L 1998 Master stability functions for synchronized coupled systems *Phys. Rev. Lett.* **80** 2109–12
[36] Thomas R 1999 Deterministic chaos seen in terms of feedback circuits: analysis, synthesis, 'labyrinth chaos' *Int. J. Bifurcation Chaos* **9** 1889–905
[37] Sprott J C 2014 A dynamical system with a strange attractor and invariant tori *Phys. Lett. A* **378** 1361–3
[38] Dadras S and Momeni H R 2009 A novel three-dimensional autonomous chaotic system generating two, three and four-scroll attractors *Phys. Lett. A* **373** 3673–42
[39] Rössler O E 1976 An equation for continuous chaos *Phys. Lett. A* **57** 397–398
[40] Sprott J C 2010 *Elegant Chaos* (World Scientific Publishing)
[41] Lorenz E N, Shlesinger M F and Zachary W 1986 Atmospheric models as dynamical systems *Perspectives in Nonlinear Dynamics: 28 - 30 May 1985 Naval Surface Weapons Center* ed R Cawley and A W Saenz (World Scientific Publishing)
[42] Pan L, Zhou W, Fang J and Li D 2010 A new three-scroll unified chaotic system coined *Int. J. Nonlinear Sci.* **10** 462–74
[43] Lorenz E N 2006 Predictability — a problem partly solved *Predictability of Weather and Climate* ed T Palmer and R Hagedorn (Cambridge University Press)
[44] Kieser R, Reynisson P and Mulligan T J 2005 Definition of signal-to-noise ratio and its critical role in split-beam measurements *ICES J. Mar. Sci.* **62** 123–30
[45] Luckinbill L S 1973 Coexistence in laboratory population of Paruinecium aurelia and its predator Didinium nasutum *Ecology* **54** 1320–7
[46] Rosenblum M G, Pikovsky A S and Kurths J 1996 Phase synchronization of chaotic oscillators *Phys. Rev. Lett.* **76** 1804–7