IMPROVED PARAMETER ESTIMATION FROM NOISY TIME SERIES FOR NONLINEAR DYNAMICAL SYSTEMS

TOMOMICHI NAKAMURA*,†, YOSHITO HIRATA‡, KEVIN JUDD§
and DEVIN KILMINSTER
Centre for Applied Dynamics and Optimization,
School of Mathematics and Statistics,
The University of Western Australia, 35 Stirling Hwy, Crawley,
WA 6009, Australia
†tomo@maths.uwa.edu.au
‡kevin@maths.uwa.edu.au

MICHAELE SMALL
*Department of Electronic and Information Engineering,
The Hong Kong Polytechnic University, Hung Hom,
Kowloon, Hong Kong
†Institute of Industrial Science, The University of Tokyo,
4-6-1 Komaba, Meguro-ku, Tokyo 153-8505, Japan

Received March 21, 2006; Revised April 18, 2006

In this paper we consider the problem of estimating the parameters of a nonlinear dynamical system given a finite time series of observations that are contaminated by observational noise. The least squares method is a standard method for parameter estimation, but for nonlinear dynamical systems it is well known that the least squares method can result in biased estimates, especially when the noise is significant relative to the nonlinearity. In this paper, it is demonstrated that by combining nonlinear noise reduction and least squares parameter fitting it is possible to obtain more accurate parameter estimates.

Keywords: Gradient descent; parameter estimation; state estimation; the least squares method.

1. Parameter Estimation in a Perfect Model Class

We consider the problem of estimating the parameters \( \lambda \in \mathbb{R}^k \) in a model \( x_{t+1} = f(x_t, \lambda), x_t \in \mathbb{R}^d \); of a nonlinear deterministic dynamical system given only a finite time series of noise contaminated observations \( \{s_t\}_{t=1}^n \) of the states \( \{x_t\}_{t=1}^n \). In order to avoid certain technical issues that detract from the core ideas of our exposition, it will be assumed that the observations measure all the state variables directly, and that the noise contamination is an independent additive isotropic Gaussian noise. That is, \( s_t = x_t + \epsilon_t \) for each \( t \), where the \( \{\epsilon_t\}_{t=1}^n \) are independent Gaussian variates with each component of mean zero and fixed variance. In the Appendix we deal with the case of scalar observations.

To avoid certain deeper technical issues we will assume that we have a perfect model class, that is,
there is a value of $\lambda$ where the model $x_{t+1} = f(x_t, \lambda)$ is identical to the system. Although the perfect model class assumption is never realized in practice, it is the only context in which parameter estimation is philosophically meaningful. If the model class is imperfect, then there is no correct value of $\lambda$, although there may be some sense optimal values of $\lambda$. To avoid difficulties of stating exactly what optimal parameter values should mean in an imperfect model class, we simply restrict attention to the perfect model class for the purposes of this paper. However, we can easily apply this method to actual examples.

Estimation of the parameters of dynamical systems is a very old problem, which has its roots in the work of Laplace, Lagrange and Gauss [Stigler, 1986]. Recently there has been renewed interest in parameter estimation of deterministic models, driven partly by rather dogmatic assertions from adherents to Bayesian methods [Meyer & Christensen, 2001], and the more pragmatic concerns of physicists [McSharry & Smith, 1999; Judd, 2003; Smirnov et al., 2002; Smirnov et al., 2005; Voss et al., 2004; Pisarenko & Sornette, 2004; Smelyanskiy et al., 2005]. The aim of this paper is to present a new algorithm for parameter estimation of deterministic models. The key feature of the new algorithm is its combination of standard methods for estimating states with standard methods for estimating parameters. The specific methods employed to estimate parameters and states are not the novelty, but the combination of state estimation with parameter estimation improves the accuracy of parameter estimation.

In general terms the sequence of states $\{x_t\}_{t=1}^n$, and hence the sequence of observations $\{s_t\}_{t=1}^n$, is determined by the parameters $\lambda$, so consequently, the estimation of the parameters from the observations is an inversion problem. If the model $f$ is linear, then, under our assumptions on the noise, the inversion problem is solved by a projection of a pseudo-inverse. However, when the model $f$ is nonlinear, the dependence of state on parameters is nontrivial and the inversion problem is also nontrivial, even when there is no noise on the observations. The presence of noise in the observations creates an even more difficult problem.

In this paper, we attempt to solve the inversion problem by an iterative algorithm that alternates between estimation of the parameters and estimation of the states. We will employ standard methods for each estimation process. To estimate the parameters of given states we will use least squares parameter estimation (LSPE). Estimation of the states is often called filtering and there are a number of standard methods available, for example, the extended Kalman filter [Walker & Mees, 1997, 1998] or gradient descent noise reduction, also called gradient descent state estimation (GDSE) [Kostelich & Schreiber, 1993; Judd & Smith, 2001, 2004; Ridout & Judd, 2001], we will use GDSE.

1.1. Least squares parameter estimation (LSPE)

It is common to estimate the parameters $\lambda$ of a model $x_{t+1} = f(x_t, \lambda)$ using least squares parameter estimation (LSPE). If the observations $\{s_t\}_{t=1}^n$ are of the entire state, then LSPE requires solving the optimization problem

$$
\min_{\lambda} \sum_{t=1}^{n-1} \|s_{t+1} - f(s_t, \lambda)\|^2.
$$

This method implicitly assumes that the noise is Gaussian, independent, identical and isotropic for each observation, so that $s_t = x_t + \epsilon_t$, $\epsilon_t \sim N(0, \sigma^2)$. For nonlinear models LSPE is well known to given estimates that can have significant bias, especially when the noise is not small. This bias can be attributed to the so-called “error in variables” problem [McSharry & Smith, 1999], that is, LSPE effectively assumes noise only affects the “response” variables $s_{t+1}$ and not the “regressor” variables $s_t$, but this is clearly not the case.

The parameter estimates would be much less biased if we could solve the optimization problem

$$
\min_{\lambda} \sum_{t=1}^{n-1} \|s_{t+1} - f(x_t, \lambda)\|^2,
$$

where $x_t$ is the true state at time $t$. The ideal situation would be to solve the optimization problem using only the true states

$$
\min_{\lambda} \sum_{t=1}^{n-1} \|x_{t+1} - f(x_t, \lambda)\|^2.
$$

Of course, we cannot know the true states $\{x_t\}_{t=1}^n$, so in Eq. (1) the observations $s_t$ is being used as a proxy for the true state $x_t$.

The key idea of this paper is that by employing some kind of filtering or state estimation technique
we might be able to obtain a better proxy of the true state than the raw observation $s_t$.

1.2. Gradient descent state estimation (GDSE)

Assume in our perfect model class that $f: \mathbb{R}^d \to \mathbb{R}^d$ is a diffeomorphism, that is, differentiable with differentiable inverse. Let $\{x_t \in \mathbb{R}^d\}_{t=1}^n$ be an arbitrary sequence of states, and regard this sequence of states as a vector in $\mathbb{R}^{nd}$, that is, $x = (x_1, x_2, \ldots, x_n) \in \mathbb{R}^{nd}$. Also let $s = (s_1, s_2, \ldots, s_n)$ denote a time series of observations of $x$, and define $\delta_t \in \mathbb{R}^d$ so that $s_{t+1} - \delta_{t+1} - f(s_t - \delta_t) = 0$. For ease of exposition we have assumed that the observation $s_t$ is of the entire state $x_t$. Often all the state variables are not observed so in the Appendix we discuss the case of scalar time series observations and a state space defined by time-delay embedding.

Define the indeterminism $L: \mathbb{R}^{nd} \times \mathbb{R}^d \to \mathbb{R}$ of a sequence of states $x$ by

$$L(x, \lambda) = \frac{1}{2} \sum_{t=1}^{n-1} \|x_{t+1} - f(x_t, \lambda)\|^2.$$  \hspace{1cm} (4)

It should be noted that $L(x, \lambda) = 0$ precisely when the states $x_t \in \mathbb{R}^d$ form a deterministic trajectory of the map $f(\cdot, \lambda)$. For more details on the meaning of indeterminism see [Judd et al., 2004b]. Generally, noisy observations do not result in deterministic trajectories. GDSE assumes that a trajectory close to the observations can be obtained by minimizing $L$ through gradient descent, using the observations $s$ as the starting point. This is achieved by solving the differential equations

$$\dot{x}(\tau) = -\nabla L(x(\tau)) = -\frac{\partial L}{\partial x},$$  \hspace{1cm} (5)

with $x(0) = s$ and finding the limit of $x(\tau)$ as $\tau \to \infty$ [Ridout & Judd, 2001]. Note the distinction between gradient descent convergence time $\tau$ and model time $t$. Writing out the gradient in Eq. (5) explicitly for each component we have

$$\frac{\partial L}{\partial x_t} = \begin{cases} -df(x_1)^T (x_2 - f(x_1)), & t = 1 \\ (x_t - f(x_{t-1})) & 1 < t < n \\ (x_t - f(x_{t-1})), & t = n \end{cases}$$  \hspace{1cm} (6)

where $T$ indicates the transpose and $df(x_t)$ is the Jacobian derivative of $f$ at $x_t$. More details about the properties of this gradient descent method of noise reduction can be found in [Judd & Smith, 2001; Ridout & Judd, 2001].

There are a number of important questions that need to be addressed concerning the convergence of GDSE. The formulation of GDSE stated thus far requires solving an ordinary differential equation, Eq. (5), until convergence is achieved. In practice, an appropriate integration scheme is required and an appropriate stopping criterion must be chosen. Both of these can significantly effect the computation time and accuracy of results. It should be noted, however, that it is not always necessary to achieve complete convergence, particularly in the early stages of the iterative algorithm we present later.

1.2.1. Integration method: Euler approximation

Equation (5) is an ordinary differential equation, which can be solved by standard integration packages. Ideally the integration should employ an adaptive step size and be able to cope with stiff systems of equations [Press et al., 1995; Shampine & Reichlet, 1995]. On the other hand, in the algorithm presented in the next section, it may not be warranted to go to these lengths in the early stages of the algorithm, or when the data has a large amount of noise, or when very precise estimation is not required.

The simplest integration of Eq. (5) is achieved by the Euler step method [Judd et al., 2004b]. In this case GDSE reduces to simple iterative algorithm

$$x_t \to x_t - \Delta \begin{cases} -df(x_1)^T (x_2 - f(x_1)), & t = 1 \\ (x_{t+1} - f(x_t)) - df(x_t)^T (x_t - f(x_{t-1})), & 1 < t < n \\ (x_t - f(x_{t-1})), & t = n \end{cases}$$  \hspace{1cm} (7)

where $\Delta$ is a suitable step size. In the latter experiments, we use this Euler integration scheme throughout with $\Delta = 0.1$.

1.2.2. Stopping criterion for GDSE

A number of different stopping criteria are possible, including: a fixed integration period or a fixed number of iterates of the Euler integration Eq. (7); a lower bound on the convergence rate, for example, when the relative change in indeterminism in Euler integration step is smaller than some threshold; when difference between the indeterminism
of the data $I(s, \lambda)$ and the current sequence of states $I(x, \lambda)$ is less than some threshold.

In the following experiments we used the Euler integration method and stopped when the difference of absolute indeterminism value between previous data and current data of $x$ in consecutive iterations is smaller than $10^{-5}$.

2. Combined State and Parameter Estimation

Our claim is that we can better estimate parameters if we can obtain better proxy estimates for the true states. The catch is that in order to employ state estimation one needs to know the correct parameter values [Heller, 1962]. We propose using an iterative algorithm where we estimate parameters using the most recent estimates of the states, then re-estimate the states using the most recent estimates of the parameters. As might be expected, the actual algorithm proposed is a little more subtle than this.

We adopt the following notation for a fixed model class $f(x, \lambda)$:

- $\lambda' = \text{least sq}(x) = \arg\min_{\lambda} L(x, \lambda)$, means that $\lambda'$ is the least squares solution to the parameter estimation problem, given data $x$.
- $x' = \text{filter}(x, \lambda)$, means that $x'$ is the sequence of state estimates obtained from a sequence of states $x$ given parameters $\lambda$. In our implementation the filter is gradient descent noise reduction.

2.1. The algorithm

Choose and fix scalars $\alpha, \beta \in (0, 1)$ and an integer $N > 0$. The purpose of $\alpha, \beta$ and $N$ we explain after stating the algorithm.

0. Set $x_0 = s$, $\lambda_0 = \text{least sq}(s)$, $k = 0$
1. Compute $L_k = L(x_k, \lambda_k)$. Stop if $L_k$ attains a minimum (see note on stopping criterion below).
2. Set $x'_k = \text{filter}(x_k, \lambda_k)$.
3. Set $x_{k+1} = (1 - \alpha)x_k + \alpha x'_k$.
4. $\lambda'_k = \text{least sq}(x_{k+1})$.
5. $\lambda_{k+1} = (1 - \beta)\lambda_k + \beta \lambda'_k$.
6. Increment $k$ and return to step 1.

The algorithm proceeds as follows. In step 2 GDSE obtains filtered data $x'_k$ from the data $x_k$ and parameter value $\lambda_k$. The original and filtered data are blended in step 3 using the proportionality factor $\alpha$. Then new parameters $\lambda'_k$ are estimated from the blended data in step 4. The new parameters $\lambda'_k$ obtained in step 4 are blended with the previously used parameter $\lambda_k$ using the proportionality factor $\beta$ in step 5.

Note that we do not set $x_{k+1}$ and $\lambda_{k+1}$ to the newly computed estimates $x'_k$ and $\lambda'_k$, but rather just move the current estimates towards these newly computed values. This is necessary to maintain stability because the initial estimates might be poor. The parameters $\alpha$ and $\beta$ limit the rate of convergence. Smaller values of $\alpha$ and $\beta$ generally provide more stability, especially in the initial iterations, but smaller values will increase the calculation time. The parameter $\alpha$ is required to avoid rapid convergence to an incorrect local minimum. However, even so, we expect that the updated state $x_{k+1}$ may be biased. To avoid inheriting that bias in $\lambda_{k+1}$ we also include the parameter $\beta$. This parameter helps to ensure that the estimated model parameters $\lambda_{k+1}$ are not overly influenced by bias in $x_{k+1}$.

2.2. Stopping criteria

Provided $\alpha$ and $\beta$ are chosen to maintain stability, the stated algorithm will converge to states $x$ and parameters $\lambda$ where the indeterminism $L$ attains a minimum. The minimum value that can be obtained is zero, which corresponds to $x$ being the trajectory of the model for the given $\lambda$. Consequently, the stopping criteria of the algorithm is the minimization of $L(x_k, \lambda_k)$. This is not necessarily the best stopping criteria. In fact, it does not guarantee the algorithm will stop, because $L(x_k, \lambda_k)$ could just decrease monotonically. It was, however, found to be adequate in the experiments we describe.

2.3. Multiple applications of algorithm

The parameter estimates of the stated algorithm may be further improved by repeated application of the algorithm where the initialization in step 0 set $\lambda_0$ to the parameter values estimated in the previous iteration. This can provide improvements because the initial parameter estimates from LSPE may be far from the correct values, especially when noise level is high.

3. Examples

We consider two nonlinear systems: the Henon map [1976] and the Ikeda map [1979].

In the case of the Henon map we choose as the perfect model class second-order difference maps

$$x_t = A_0 + A_1 x_{t-2} + A_2 x_{t-1}^2,$$  \hfill (8)
where the perfect model (system) has \( A_0 = 1.0, A_1 = 0.3 \) and \( A_2 = -1.4 \). Often the Henon system is thought of as a two-dimensional system, and hence one can consider our experiment here as an example of incomplete observation of the state and use of a time-delay embedding model as described in the Appendix.

For the Ikeda map we choose the perfect model class,

\[
f(x, y) = \left( \frac{1 + \mu (x \cos \theta - y \sin \theta)}{\mu (x \sin \theta + y \cos \theta)} \right), \quad (9)
\]

where the perfect model (system) has \( \theta = a - (b/(1 + x^2 + y^2)) \) with \( \mu = 0.83 \), \( a = 0.4 \) and \( b = 6.0 \). In the Ikeda map we allow observations of the entire state.

In all the experiments we have used sequences of 1000 observations and Gaussian observational noise. We will express the noise level in terms of the approximate signal to noise ratio (SNR), measured in decibels. We demonstrate with applications to three observational noise levels, 40 dB as relative small noise level, 20 dB as moderate noise level and 10 dB as large noise level. We mostly present detailed results only for the 20 dB case, because the behavior of 10 dB and 40 dB cases are basically similar.

### 3.1. Initial least squares estimates

Table 1 shows the initial parameter estimates \( \lambda_0 \) using LSPE for the Henon and Ikeda maps for various noise levels. As the noise level increases (that is, as the value of dB decreases), the parameter estimates deviate more from the correct values. It should be noted from Eq. (9) that we need a nonlinear least squares method to estimate the parameters (that is, \( \mu, a \) and \( b \) in the Ikeda map. We have used MATLAB’s \texttt{lsqcurvefit} routine.\(^1\)

### 3.2. First application of algorithm

First we examine parameter estimates obtained from one application of the algorithm. In all the experiments we have chosen \( \alpha = \beta = 0.5 \).

Figure 1 shows how the parameter estimates \( \lambda_k \) develop for the Henon and Ikeda experiments. In both experiments there is an initial rapid movement followed by slower convergence. In these experiments, the convergence is monotonic. Except for Fig. 1(b) the monotonic movement is always toward the correct parameter values; in Fig. 1(b), however, the parameter \( A_1 \) passes through the correct value and converges to a slightly higher value.

Figure 2 shows the development of the indeterminism. In both experiments, the indeterminism is converging, but apparently not to zero. It is desirable that the indeterminism converges to zero, but for technical reasons for likely nonhyperbolicity of the maps, it should be expected that the convergence will become arbitrarily slow, and hence appear to converge to a nonzero value [Ridout & Judd, 2001; Judd, 2006].

Figure 3 shows plots of the original 20 dB noise observations and the final state estimates of the algorithm. In the case of the Henon data, Figs. 3(a) and 3(b), show time-delay embeddings.\(^2\)

---

\( ^1 \text{lsqcurvefit} \) is one of the functions in MATLAB’s Optimization Toolbox. It does nonlinear least squares and curve approximation.

\( ^2 \)There are two points that are slightly off the attractor in Fig. 3(b), where these points are around \((0.918, -0.647)\) and \((-0.424, 0.918)\). These data are the first and second points of the Henon map data. It is well known that any state estimation method, including gradient state estimation, is expected to have slightly larger errors for states at the beginning and end of the time series [Judd & Smith, 2001]. For those states at the beginning, this error is off the attractor, and hence visible in these figures.
Fig. 1. Behavior of the parameter estimates for first application of the algorithm to the Henon and the Ikeda maps. For the Henon map, (a) $A_0$, (b) $A_1$ and (c) $A_2$. For the Ikeda map, (d) $\mu$, (e) $a$ and (f) $b$. The dotted line is the correct value.

Fig. 2. Behavior for the Henon and the Ikeda maps of the indeterminism $L_k$ during the first application of the algorithm. We note that the Y-axis is shown on a logarithmic scale: (a) for the Henon map, and (b) for the Ikeda map.
3.3. Repeated application of algorithm

We now examine the effect of repeated application of the basic algorithm, that is, at the completion of the algorithm it is restarted using the previous parameter estimates for $\lambda_0$, rather than using the LSPE values. We repeat the application ten times to examine the convergence of repeated application of the algorithm. Figures 4 and 5 show the parameter estimates and indeterminism for repeated application to the Henon and Ikeda experiments, respectively. Both figures show that the first application provides the largest improvement, and there was a monotonic convergence after the second application.

Table 2 shows the comparison of parameter estimates for the Henon and Ikeda experiments.
Fig. 4. Behavior of the parameter estimates for repeated application of the algorithm to the Henon and Ikeda maps, where observational noise level is 20 dB. The × indicates estimated parameters and the dotted line is the correct value: (a)–(c) for the Henon map, and (d)–(f) for the Ikeda map.

Fig. 5. Behavior of the indeterminism for the repeated application of the algorithm to the Henon and Ikeda maps, where observational noise level is 20 dB. We note that the Y-axis is shown on a logarithmic scale. The × indicates the indeterminism: (a) for the Henon map and (b) for the Ikeda map.
Table 2. Parameter estimates after each application of the algorithm for the Henon and Ikeda maps with 10 dB, 20 dB and 40 dB noise levels. The LSPE is included for comparison.

| SNR 10 dB  | Henon map |  |  | Ikeda map |  |  |
|------------|-----------|  |  |   |  |  |
|            | \( A_0 \) | \( A_1 \) | \( A_2 \) | \( \mu \) | \( a \) | \( b \) |
| LSPE       | 0.745548  | 0.250284 | -0.885106 | 0.735939 | -0.049027 | 5.062187 |
| 1st        | 0.516021  | 0.321725 | -0.660197 | 0.832149 | 0.346155  | 5.847787 |
| 5st        | 0.366387  | 0.194475 | -0.392020 | 0.829681 | 0.366609  | 5.949808 |
| 10st       | 0.353430  | 0.195844 | -0.410225 | 0.827892 | 0.371472  | 5.970428 |
| SNR 20 dB  |  |  |   |  |  |  |
| LSPE       | 0.957543  | 0.293912 | -1.319513 | 0.816888 | 0.352389  | 5.910301 |
| 1st        | 1.002034  | 0.300781 | -1.385922 | 0.826931 | 0.387994  | 5.986636 |
| 5st        | 0.999415  | 0.300243 | -1.390872 | 0.826979 | 0.396319  | 6.004287 |
| 10th       | 0.998737  | 0.300227 | -1.391822 | 0.826830 | 0.397357  | 6.006573 |
| SNR 40 dB  |  |  |   |  |  |  |
| LSPE       | 0.998779  | 0.299903 | -1.397379 | 0.829588 | 0.399862  | 6.000289 |
| 1st        | 0.999120  | 0.299952 | -1.398904 | 0.829654 | 0.400109  | 6.001395 |
| 5th        | 0.999470  | 0.300009 | -1.399474 | 0.829717 | 0.400374  | 6.001942 |
| 10th       | 0.999492  | 0.300015 | -1.399524 | 0.829721 | 0.400400  | 6.001990 |
| Correct    | 1.0       | 0.3     | -1.4     | 0.83     | 0.4       | 6.0       |

Fig. 6. Reconstructed Henon maps using contaminated data and data when the indeterminism for the first application of the algorithm converges: (a) data contaminated by 10 dB observational noise, and (b) the data when the indeterminism converges.

It can be seen that the algorithm provided more accurate parameter estimates in all cases except the Henon map when the noise level was 10 dB; we discuss this failure in more detail in the next section. It might be also noted from Table 2 that when the algorithm was successful it provides more improvement relative to the initial LSPE, for the larger noise levels of 10 dB and 20 dB than the smaller noise level of 40 dB. This is almost certainly because at higher noise levels the observations are not a good proxy for the true states, and so the benefits of the obtained better proxy are significant.

3.4. A failure of the algorithm and future work

It has been noted in the previous two sections that the algorithm failed to find more accurate parameters estimates than LSPE in the Henon system.
when observational noise was at the 10 dB level. Figure 6 shows time-delay embedding of the original data and the algorithm’s final state estimates, where the result is obtained by the first application of the algorithm. It is clear that the algorithm has converged to a model with a stable fixed point. Although this might be a reasonable model when the noise is very large (and therefore completely masking the dynamics), in this case the dynamics are not completely masked.

Consequently, we conclude that the algorithm is not reliable for very large noise levels, but on the other hand, it could be improved. For example, the algorithm takes no note of the residuals of the data given the filtered states. In the failure we have observed these residuals increased significantly from residuals of the initial LSPE. Incorporating this residual information in the algorithm may prevent the failure. It could have also been avoided simply by using smaller $\alpha$ and $\beta$. We need to investigate this further.

4. Conclusion

We have described an algorithm for obtaining parameter estimates of deterministic systems that improve upon estimates obtained by applying least squares methods. The proposed method combines least squares parameter estimation with nonlinear state estimation, in particular a gradient descent technique. We have applied the method to two nonlinear systems, both with significant observational noise. In most cases, the parameter values obtained were more accurate than those obtained by least squares. A failure was noted, however, with large noise levels, and so improvements to the algorithm should be sought.

Acknowledgment

This work was partially supported with a direct allocation from the Hong Kong Polytechnic University (A-PG58).

References


Appendix

In this appendix, the gradient descent method for estimating states is modified for use in scalar time series. In particular, it is proved that there exists a strict Lyapunov function for this implementation. The gradient descent method is sometimes used for estimating states in dynamical systems [Judd & Smith, 2001; Ridout & Judd, 2001]. However, so far it has only been formulated for full state models.

Suppose that one has a time series \( x = \{x_t \in \mathbb{R}\}_{t=1}^{n} \). Let \( M = \mathbb{R}^d \) be the \( d \)-dimensional time-delay embedding space, that is, define \( X_t = (x_{t-d+1}, x_{t-d+2}, \ldots, x_t) \in M \). Let \( f : M \times \mathbb{R}^k \to \mathbb{R} \) be a model for the scalar time series, that is, \( f(X_t, \lambda) \) forecasts the scalar \( x_{t+1} \). Define

\[
L(x) = \frac{1}{2} \sum_{t=d}^{n-1} \|x_{t+1} - f(X_t, \lambda)\|^2.
\]

This quantity \( L(x) \) is non-negative, achieving the minimum 0 when \( x \) is a trajectory for a given \( \lambda \). Now consider gradient descent of \( L(x) \). Define the following ordinary differential equation:

\[
\frac{dx}{d\tau} = -\frac{\partial L}{\partial x}, \quad x(0) = s.
\]

Writing out the components explicitly one obtains the following, where we stop indicating the dependence of \( f \) on \( \lambda \),

\[
\frac{dx}{d\tau} = \begin{cases}
\sum_{i=d-t}^{d-1} \frac{\partial f(X_{t+i})}{\partial x_t} (x_{t+i+1} - f(X_{t+i})), & 1 \leq t \leq d \\
-(x_t - f(X_{t-1})) + \sum_{i=0}^{d-1} \frac{\partial f(X_{t+i})}{\partial x_t} (x_{t+i+1} - f(X_{t+i})), & d < t \leq n-d \\
-(x_t - f(X_{t-1})) + \sum_{i=0}^{n+1-t} \frac{\partial f(X_{t+i})}{\partial x_t} (x_{t+i+1} - f(X_{t+i})), & n-d < t \leq n \\
-(x_t - f(X_{t-1})). & t = n
\end{cases}
\]

We prove the following theorem:

**Theorem 1.** \( L(x) \) is a strict Lyapunov function for \( x \in \mathbb{R}^n \).

**Proof.** Letting \( \Omega \) be some neighborhood of \( \tilde{x} \), a function \( L(x) \) is called a strict Lyapunov function [Alligood et al., 1997, p. 305], if

- \( L(\tilde{x}) = 0 \) and \( L(x) > 0 \) for all \( x \neq \tilde{x} \) in \( \Omega \), and
- \( dL(\tilde{x})/d\tau = 0 \) and \( dL(x)/d\tau < 0 \) for all \( x \neq \tilde{x} \) in \( \Omega \).

As \( L(x) \) is non-negative and is zero at a desired solution, we just have to show the condition for \( dL(x)/d\tau \).

Define \( g(x) \) in the following way:

\[
g(x) = \begin{pmatrix}
x_{d+1} - f(X_d) \\
x_{d+2} - f(X_{d+1}) \\
\vdots \\
x_{t+1} - f(X_t) \\
\vdots \\
x_n - f(X_{n-1})
\end{pmatrix}.
\]

Observe that \( g(x) = 0 \) for trajectory. Then \( L(x) \) is given by

\[
L(x) = \frac{1}{2} g(x)^T g(x).
\]
Therefore $dx/d\tau$ can be written in the following form:

$$\frac{dx}{d\tau} = -\left(\frac{\partial g(x)}{\partial x}\right)^T g(x).$$

It is straightforward that one has

$$\frac{dL(x)}{d\tau} = \frac{dg(x)^T}{d\tau} g(x) = \left(\frac{\partial g(x)}{\partial x} \frac{dx}{d\tau}\right)^T g(x)$$

$$= \frac{dx^T}{d\tau} \left(\frac{\partial g(x)}{\partial x}\right)^T g(x) = \frac{dx^T}{d\tau} \left(-\frac{dx}{d\tau}\right)$$

$$= -\left\|\frac{dx}{d\tau}\right\|^2 \leq 0,$$

resulting in $dL(x)/d\tau \leq 0$.

Lastly we consider the case of equality. Observe that $dL(x)/d\tau = 0$ if and only if $dx/d\tau = 0$. If $g(x) = 0$, then $dx/d\tau = 0$. Conversely if $dx/d\tau = 0$, it holds that $x_n = f(X_{n-1})$. Then applying back-substitutions recursively also gives $x_i = f(X_{i-1})$ for $i = d + 2, d + 3, \ldots, n - 1$, leading to $g(x) = 0$. Therefore, one can conclude that $dL(x)/d\tau = 0$, if and only if $g(x) = 0$.  

A strict Lyapunov function guarantees the convergence of the gradient descent method to a trajectory [Alligood et al., 1997, p. 307].