APPROXIMATION AND COMPRESSION OF ARBITRARY TIME-SERIES BASED ON NONLINEAR DYNAMICS

Maciej J. Ogorzalek

Instituto de Microelectronica de Sevilla
Centro Nacional de Microelectronica
Avda. Reina Mercedes s/n. 41012 Sevilla, Spain

ABSTRACT

Given a measured output signal from a nonlinear deterministic process in the form of a time series we estimate initial conditions in successive time intervals which reproduce in an optimal way sections of the output signal. Thus the measured signal can be coded in terms of model equations, its parameters and a set of initial conditions. We achieve signal approximation and compression at the same time. We discuss continuous-time and discrete-time approaches.

1. PROBLEM STATEMENT

Data acquisition, processing and storage are of paramount importance in a variety of real life problems [4]. Meteorology, astronomy, seismology, physiology, medical observations, solid-state physics, electronic circuits, telecommunication channels are just a few examples [1], [6].

We consider the following problem: Suppose we have a complicated time-waveform measured from some real system eg. ECG, EEG, speech or any other. Storage of such signals is a difficult problem as to maintain the accuracy the signal must be sampled at a high frequency and in a digital form requires large word-length. Suppose we have the signal measured during some time interval T (e.g. as shown in Fig.1). Let us divide the interval T in an arbitrary way into sub-intervals (equal or not) - this can be done by sampling the signal by an A/D converter or introduced artificially depending on application. In this way we define a sequence of points. The idea is to store/code the waveform using these chosen points only. We can distinguish two approaches:

- Knowing the pieces of waveforms find initial conditions for each interval assuming that the output has been produced by a continuous-time nonlinear deterministic system. Such kind of coding can be effective only in the case when the decoder could reproduce the pieces of waveform between the samples.
- Knowing the sequence of points find the initial condition for a discrete-time nonlinear deterministic system (one- or higher-dimensional map) which with a sufficient accuracy can be given sequence of points.

In the second case we would certainly need a finer sampling as the system will not be able to produce intermediate points between the samples. In the first case one can expect that the generator can reproduce any intermediate points between the samples. In the nonlinear dynamics literature closely related problems known as shadowing and noise removal have also been considered [1].

2. CONTINUOUS-TIME APPROACH

In our earlier studies we considered a continuous-time dynamical system of the Lur’e form [7]:

\[ \dot{x}(t) = Ax(t) + Bu(t) \]
\[ y(t) = C^T x(t) \]
\[ u(t) = f[g(t)] \]

where: \( x(t) \), \( B \), \( C \) \( \in \mathbb{R}^n \), \( y(t) \), \( u(t) \) \( \in \mathbb{R} \), \( A \) - \( n \times n \) real matrix. \( f : \mathbb{R} \rightarrow \mathbb{R} \) is a locally Lipschitz function for \( t \geq 0 \). Suppose we observe a transient signal of finite length \( T \) which is the output of some system in a time interval \([t_0, t_0 + T]\). Let us suppose further that the trajectory associated with the signal \{\( g(t) \)\} begins with an initial state \( x_0 = x(t_0) \) and ends at a final state \( x_T = x(t_0 + T) \).

2.1. Solution I.

For the Lur’e type of systems it is possible to write explicitly the equation for the output signal:

\[ y(t) = C^T e^{A(t-t_0)} x_0 + C^T \int_{t_0}^{t} e^{A(t-	au)} B f[g(\tau)] d\tau \]

Figure 1: Cutting of the original waveform into sections to be "chaos-coded".
Hence in a simplified form we can write:

\[ C^T e \cdot A^i x_0 = y(t) - (g \ast f)(t) \]  

(3)

where \( y \) is the impulse response of the linear part of the system and \( \ast \) denotes the convolution.

Equation (3) is solvable iff \( y - g \ast f \) belongs to the range of the observability map \( \mathbf{P} : R^n \ni x_0 \rightarrow \mathbf{P} x_0 \in L^2(t_0, t_4) \) where \( (\mathbf{P} x_0)(t) = C^T e \cdot A^i x_0 \).

Let \( G \) be the Gram matrix of this systems and \( q \) be the vector of the scalar products (in \( L^2(t_0, t_4) \)) of this system with the right-hand side of (3). Then using the orthogonal projection theorem \([7]\) one gets

\[ G x_0 = q \]  

(4)

which can be solved using the pseudo-inverse concept giving the best least-square approximation of \( x_0 \) \([3]\).

2.2. Solution II.

Let us consider next a more general class of systems described by:

\[ \frac{dx}{dt} = F(x) \]  

(5)

The problem to be solved is to find \( x(0) \) (vector of initial states) for the equation describing the time evolution knowing the sampled version of the scalar output signal \( y(t) \).

We can use an adaptive procedure (originally proposed by Manevhte and Antrikar \([8]\) for the purpose of synchronization of two chaotic systems). Here we propose its use for signal coding.

For simplicity we will assume that the measured (known) signal is the first component \( x_1(t) \) of \( x(t) \). Let us assume that we take a random initial point \( z(0) \) and the time evolution from this initial point in our system is denoted by \( z(t) \). Let \( w(t) \) denote the difference

\[ w(t) = z(t) - x(t) \]  

(6)

We search for a solution of the above equation which would satisfy \( w(t) = 0 \) and in particular we are searching for \( w(0) = z(0) - x(0) = 0 \) !!

We will use the modified Newton-Raphson method including the evolution of the system. We can use the notation \( w(i) = w(i)(z), x(i) = x(i)(\Delta t) \) where \( \Delta t \) is the sampling time interval. In a similar way we can introduce \( x(\Delta t) = x(i(\Delta t)) \) and \( z(i) = z(i(\Delta t)) \). For \( w(1) \) we can write:

\[ w(1)(z), x(0)) = w(0) + \Delta t[F(z(0)) - F(x(0))] + o(\Delta t^2) \]  

(7)

Considering

\[ 0 = w(1)(x, x(0)) = w(1)(z(0) + \delta z(0), x(0)) = w(1)(z(0), x(0)) + (\delta y(i) J_{yg(0)} w(1)(z(0), x(0)) + o(\delta y(i))^2) \]  

(8)

where \( \delta z(0) = \delta x(0) - \delta x(0) = -\delta w(0) \) we get

\[ w(1) = w(0) + \Delta t[w(0) J_{yg(0)} F(z(0))] \]  

(9)

or in a matrix form:

\[ w(1) = (I + \Delta t J_0) w(0) = A_0 w(0) \]  

(10)

Repeating the same reasoning for successive points we get:

\[ w(n) = (I + \Delta t J_{n-1}) \cdots (I + \Delta t J_0) w(0) = A_{n-1} A_{n-2} \cdots A_1 A_0 w(0) \]  

(11)

In a D-dimensional system we need D-1 equations to determine the state vector \( x(0) \). As we measured just the first component of the signal we have to consider just the first components of \( w(i) \)

\[ w^i(1) = \sum_{j=1}^{D} A_{ij}^0 w_j(0) \]

\[ w^i(2) = \sum_{j=1}^{D} A_{ij}^1 w_j(0) \]

\[ \vdots \]

\[ w^i(D-1) = \sum_{j=1}^{D} A_{ij}^{D-1} w_j(0) \]  

(12)

The numerical procedure goes as follows:

1. set the initial point \( z(0)_{old} \) in a random way fixing its first component \( z^1(0)_{old} = z(0) \) and let the system evolve following the equations of motion for some time.

2. using the vector \( z(t) \) we can write the equations (12) and solve them for components of \( w(0) = -\delta z(0) \) (also \( \delta z(0) = 0 \)).

3. The initial guess can be improved by

\[ z(0)_{new} = z(0)_{old} + \delta z(0) \]  

(13)

Notes:

* There is no guarantee of convergence !! nor finding the proper solution.

3. DISCRETE-TIME APPROACH

Let us consider now that we have a sampled version of the measured time series:

\[ y(i), i = 0 \ldots N \]

Let us suppose next that we have a discrete-time dynamical system (map)

\[ z(i + 1) = F[z(i)], \quad z(0) = z_0 \]  

(14)

We would like to find a map \( F \) and an initial condition \( z_0 \) which would reproduce a sequence \( z(i) \) in such a way that \( z(i) - y(i) \leq \varepsilon \) (\( z(i) \approx y(i) \) in an ideal case) where \( \varepsilon \) is an assumed acceptable error bound.

Science-Fiction coding theorem

Arbitrary time-series of any length can be coded using single initial condition for a one-dimensional map.

Sketch of proof:

The reasoning is the following:

* The amplitudes of the measured signal can be rescaled to the unit interval.
* Any value of the scaled time series can be represented in a binary representation - thus instead of the original time series we can further consider a series of bits (0s and 1s) - possibly longer then the original one: This sequence can
be considered as a symbolic signature of the original time series.

- Let us consider simple Bernoulli shift map - it is a well established fact that such a map can reproduce a symbolic sequence of any length. Open question remains how to find the initial point looking backwards from the end of the binary code.

**Comments:**

There exists a trade-off between the length of the data set and the precision of the initial condition - the longer the time series the better precision (number of bits) is required for storing the initial condition item. Considering a fixed-point storage implementation Bernoulli shift applied at each iteration causes loss of one bit of information i.e. there is a loss of precision of one bit per iteration.

There exist a number of approaches for finding initial conditions for a chaotic maps given an output time series. In the experiments described below we used the simplest case of a Bernoulli shift map and the extended halving method [9] or the backward iteration approach as proposed in [2] (the detailed description of the algorithms is too long for the scope of this paper and can be found in [2] and [9]).

4. RECONSTRUCTION OF WAVEFORMS FOR CHUA'S CIRCUIT

We have generated a long time series containing samples of the output from Chua's circuit (storing $x_1$ only - voltage across the $C_1$ capacitor). In the experiments we used two time intervals of 50000 samples ($\Delta t = 10^{-5}$). In all the experiment for initial condition reconstruction the signal has been down-sampled by a factor of 50. Both the first continuous-time approach using Chua's equations and the discrete-time approach using the Bernoulli shift map give excellent results it is possible to reconstruct 50000 points on the waveform from a single initial condition stored using floating point arithmetics.

![Image of waveform reconstruction](image1.png)

Figure 2: Reconstruction of a sequence containing 50000 samples. Initial condition stored using floating point representation. The measured time-series is reconstructed equally well in both cases of continuous time approximation and discrete-time approximation.

![Image of artificial signal](image2.png)

Figure 3: Artificially generated signal and the 12 pieces reconstructed by the model (Chua's circuit). In the discrete-time case the approximation is very good - we need one point to reproduce a the sequence of 2000 points on the trajectory.

5. CODING AND COMPRESSION OF AN ARTIFICIALLY GENERATED SIGNAL

For this test we generated artificially 40000 samples of a special kind of signal it was composed of a square wave of amplitude 2.0 and frequency $f_0 = 500\,Hz$ which was added with a sinusoidal signal of frequency $f_1 = 2400\,Hz$ modulated by a sawtooth waveform of frequency $f_2$. The combined signal was low-pass filtered by a first-order filter with a cut-off frequency of 4000Hz. The maximum error has been fixed to 2.5% and precision of the initial conditions for each segment to 16 bit. The whole waveform could be stored and reconstructed using just 12 initial conditions (we had to store 36 numbers representing initial conditions for 12 intervals).

We tested also the performance of the Bernoulli shift approximator using floating point coding of the initial condition. The reproduction is excellent for a series of 2000 points (downsampled version of the measured signal).

6. CODING/COMPRESSION OF ECG SIGNAL

In the experiment the approximation error has been fixed not to exceed 2.5% and the initial conditions for each segment found using continuous-time approximation are stored using 16 bit precision. The compression ratio in this case is of 8.76 (we need to store 45 parameters at 16 bit each (initial conditions for 15 intervals) + 15 times the lengths of the intervals coded at 9 bit). Comparing the results obtained in terms of ECG compression ratios we stress here that our figures are still quite modest - typically state-of-the-art ECG compression methods give results in the range of compression ratios 12-15 (at the 500Hz sampling rate) [1].

Truly amazing result is obtained by the Bernoulli shift approximator - we are able to reproduce 500 points on the trajectory using single initial point!! thus reducing the number of stored points by a factor of 45 (15 multiplied by the 3 - the dimensionality of the system) but requiring high precision for storing the initial condition! For comparing this method's performance, see Table 1 (after [5]).
of course, our technique needs to be optimized for use in compression.

<table>
<thead>
<tr>
<th>Method</th>
<th>CR</th>
<th>$f_s$ (Hz)</th>
<th>PRD (%)</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>AZTEC</td>
<td>10.0</td>
<td>500</td>
<td>28.0</td>
<td>Poor $P$ and $T$ fidelity</td>
</tr>
<tr>
<td>TP</td>
<td>2.0</td>
<td>200</td>
<td>5.3</td>
<td>Sensitive to Sampl. rate</td>
</tr>
<tr>
<td>CORTES</td>
<td>4.8</td>
<td>200</td>
<td>7.0</td>
<td>Poor $P$ fidelity</td>
</tr>
<tr>
<td>SAPA</td>
<td>3.0</td>
<td>250</td>
<td>4.0</td>
<td>High fidelity</td>
</tr>
<tr>
<td>Entropy Coding</td>
<td>2.8</td>
<td>250</td>
<td>-</td>
<td>High fidelity</td>
</tr>
<tr>
<td>DPCM Lin Pred</td>
<td>2.5</td>
<td>250</td>
<td>-</td>
<td>High fidelity</td>
</tr>
<tr>
<td>DPCM Lin Pred + Entropy cod.</td>
<td>7.8</td>
<td>500</td>
<td>3.5</td>
<td>High fidelity</td>
</tr>
<tr>
<td>Karhunen-Loeve</td>
<td>12.0</td>
<td>250</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Adapt. Hermite Functions</td>
<td>11.6</td>
<td>360</td>
<td>-</td>
<td>info on ectopic beat</td>
</tr>
</tbody>
</table>

Table 1: Comparison of ECG data compression schemes

7. CONCLUSIONS

Nonlinear models with rich dynamics offer a potentially very useful tool for advanced signal processing - specifically signal coding and compression. The main features confirmed so far are:

- Proposed approach for finding initial conditions can be performed recursively giving an excellent approximation of a given (measured) signal by sections of trajectories of a chosen model (waveform generator). One can adapt the lengths of successive intervals accordingly given the maximum error which can be fixed by the user.
- Even if no good model were available taking a “sufficiently rich” waveform generator (e.g., a suitable chaotic oscillator like Chua’s circuit in our examples) or the Bernoulli shift map in the discrete case we have shown that arbitrary waveforms could be processed.
- The continuous-time approach intrinsically offers the possibility of finding intermediate points on the trajectory - this is not the case of discrete-time approximator. In our first approach we get data smoothing and noise removal as the procedure gives the best solution in the least squares sense.
- The measured signal is coded by a set of points in the state space (the “code” consists of the generating model equation together with its parameters and a set of initial conditions and time interval lengths).
- At no additional computational cost we obtain also signal compression - instead of storing the whole waveform one can just store the “code”.
- There exists a trade-off between the length of approximated trajectory and the accuracy of initial condition. As stated in our “Science fiction theorem” one could code any time series of any length in just one initial condition provided it were infinitely accurate (infinite wordlength!). In the case of Bernoulli shift map the interpretation of results in the case of fixed-point storage is simple - on every iterate of the map one loses one bit of accuracy.
- Knowing the generating model and initial conditions one can easily regenerate the compressed signal.
- It should be stressed that the better fidelity of the continuous-time model the better the results of approximation and compression.
- Using discrete-time approximator we lose the extrapolating feature - the system is unable to find intermediate points between the measured ones. It can just reproduce a trajectory very near the given one!

8. REFERENCES