Simultaneous amplitude and frequency noise analysis in Chua's circuit – neural network based prediction and analysis.

J.-M. Friedt[1] ⋆, O. Teytaud[2], D. Gillet[1], and M. Planat[1]
friedt@lpmo.univ-fcomte.fr, oteytaud@univ-lyon2.fr

[1] Laboratoire de Physique et Métrologie des Oscillateurs (LPMO, CNRS)
32, avenue de l'observatoire, 25044 Besançon Cedex, France
[2] Laboratoire ERIC - Université Lyon II
4, avenue Pierre Mendes France, 69676 BRON Cedex, France

Abstract. A large number of simultaneous frequency and amplitude data from an electronic chaotic circuit (Chua's circuit) have been obtained. These acquisitions are validated by plotting the bifurcation diagrams of the experimental data versus the bifurcation parameter. We introduce a topological parallel between the Colpitts oscillator and Chua's circuit, and observe similar behavior of the frequency fluctuations using the Allan deviation.

We then study the efficiency of different learning algorithms for one or many steps prediction of time series. Different strategies are compared: direct long term predictions and iterated methods. Different criteria are used and compared. Simulated "perfect" data, simulated "noisy" data, and real-world measured data are used and compared. We use many different algorithms: classical ones, like backpropagation, radial basis functions, and k-nearest neighbors, but more recent ones too, like support vector machines, already used for chaotic time series. All these methods are combined to predict the order of chaotic time series.

1 Introduction

Chua's circuit is a non-linear electronic circuit which exhibits chaotic behavior for high enough values of one of its parameters, the inductor's value in our case. We have observed that in the chaotic regime (while the signal is still pseudo-periodic, *i.e.* before the bifurcation towards the double-scroll regime) the frequency fluctuations are characterized by an Allan deviation slope equal to -1. The same kind of behavior is observed in quartz oscillators.

We have developed a tool aimed at predicting these fluctuations using neural networks, based on the hypothesis of the deterministic nature of the frequency fluctuations [14] and of the related amplitude. The phenomenology of this method may be applied in a large variety of fields for the prediction of deterministic chaotic signals embedded in noisy signals.

⋆ We would like to thank Franck Lardet-Vieudrin for designing the Colpitts oscillator, François Vernotte for the phase noise measurements on our oscillators as well as Pierre Borgnat, Olivier Michel and André Elisseef for very interesting discussions.

2 Motivation of this study

Our aim is to be able to model frequency instabilities in oscillators. Our approach, based on neural networks, is not aimed at a better understanding of the physical phenomena leading to $1/f$ noise, but to be able to predict the evolution of dynamical systems displaying such fluctuation spectra. We have thus chosen to analyse Chua's chaotic circuit as an example of a highly unstable oscillator. The conclusions can be cautiously extended to more classical oscillators such as the Colpitts oscillator used in conjunction with a quartz resonator.

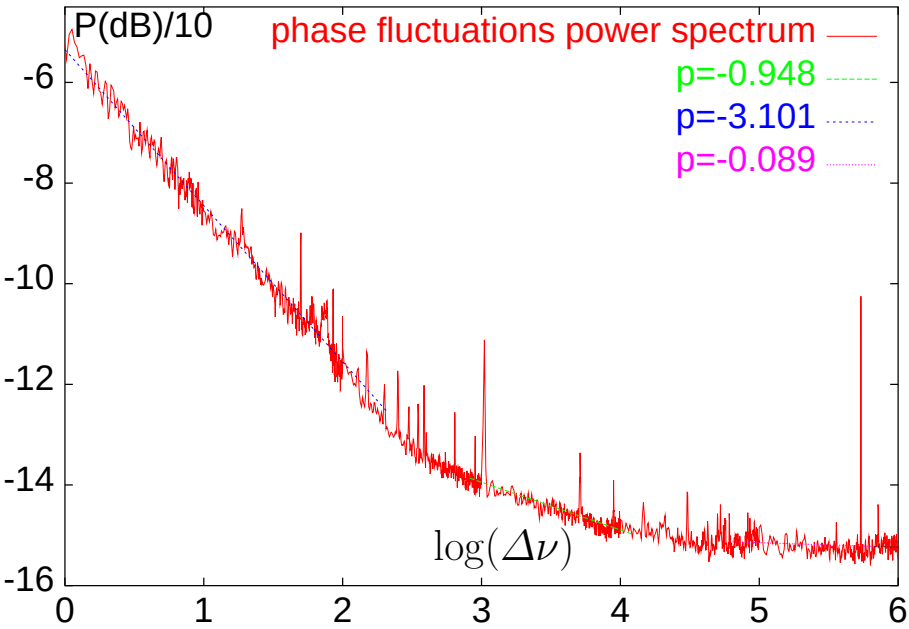

Fig. 1. Phase noise $S_\varphi(\Delta\nu)$ measured for a 11.0592 MHz resonator and linear fit from the power law expansion.

Quartz oscillators show a well known phase (and hence frequency) fluctuation behavior observed to follow a series of power law functions. The power spectral density of relative frequency fluctuations is thus $S_y(F) = h_\alpha F^\alpha$, where F is the Fourier frequency, $y = \frac{\Delta\nu}{\nu_0}$ the relative frequency shift with respect to the carrier of mean frequency ν_0 and the integer α varies from -2 to +2. An example of low frequency spectrum is provided in Fig. 1 for the case of a quartz oscillator (we used a 11.0592 MHz resonator and a finely tuned Colpitts amplifier). Here the power spectral density of phase fluctuations is $S_\varphi(F) = \frac{\nu_0}{F^2} S_y(F)$. Hence a slope equal to -3 in $S_\varphi(F)$ corresponds to $1/F$ noise in $S_y(F)$. This type of noise is still not well understood. This is one reason for attempting to

detect it in a Chua's oscillator. Instead of power spectra we will use the characterization of time dynamics in terms of Allan variance which is defined as $\sigma_y^2(\tau) = \frac{1}{2}\langle (y_{k+1}(\tau) - y_k(\tau))^2 \rangle$, with $y_k(\tau)$ the k^{th} average of samples y over the integration time τ. y is obtained by computing the average $y_\tau = \langle x_{n=[t;t+\tau]} \rangle$, x_t being our time series indexed by t and $\langle \bullet \rangle$ an average over the data sets. For a stationnary fluctuation the power spectrum and Allan variance are related. The graph of the variance versus the sampling time also obeys power laws $\sigma_\tau^2(y) = \tau^{-m}$, with $m = \alpha + 1$ (if $-2 \leq \alpha \leq 1$) and $m = 2$ (if $\alpha \geq 1$). Therefore the $1/F$ noise in $S_y(F)$ corresponds to a flicker floor $m = 0$.

A parallel between the schematics of the Colpitts oscillator and the canonical Chua's circuit [2,3] can be displayed and analyzed in order to justify our extension of the results to quartz oscillators (Fig. 2). The circuit we actually studied is the usual Chua's circuit in which L and R_2 are exchanged [1]. The non-linear differential equations used to predict the behavior of the circuit lead to chaotic oscillations after a series of bifurcations. Here we have studied simultaneously the evolution of the instantaneous amplitude and instantaneous frequency with the value L of the inductor parameter.

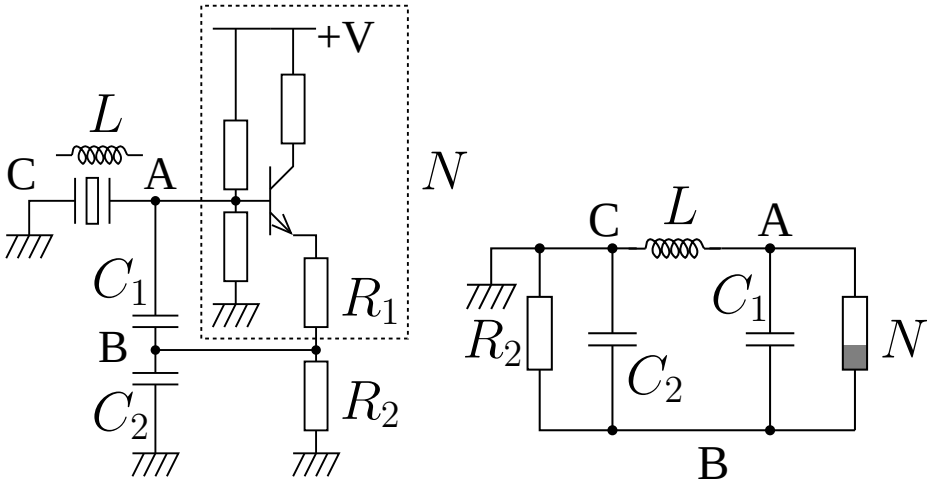

Fig. 2. Comparison between the Colpitts oscillator and Chua's circuit.

3 Experimental setup

For the realization of Chua's circuit, we used a simulated inductor described by Weldon [4]: it is less voluminous and its value is defined by a variable resistor and thus can be computer controlled thanks to the availability of computer controlled variable resistors. The non-linear negative resistor is a negative impedance converter (NIC) [15] based on an operational-amplifier [4].

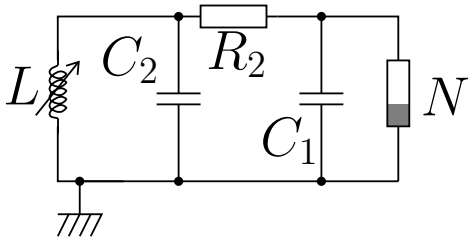

Fig. 3. Schematic of Chua's circuit we used for our data acquisitions. The parameter is the inductor's value L. Other components value are $C_2 = 1847$ nF, $C_1 = 20.3$ nF, inductor's internal resistance 3.3 Ω. R_2 is a 10 kΩ variable resistor. The inductor and the nonlinear elements are built as described by Weldon [4]. The variables measured are the frequency and amplitude of the voltage at C_2.

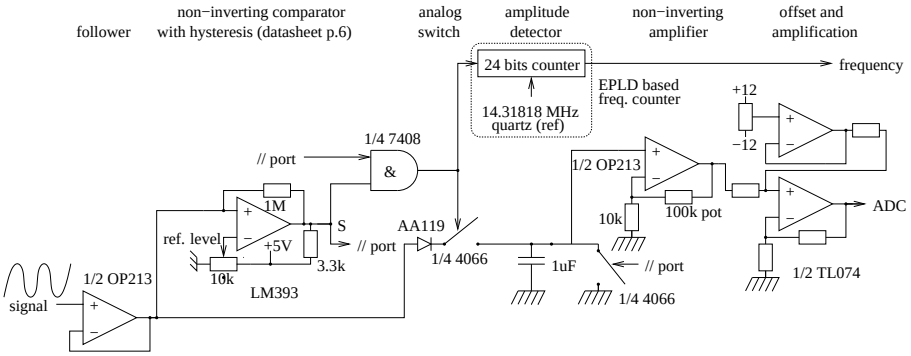

Fig. 4. Experimental setup used for measuring the frequency and amplitude of successive oscillations of Chua's circuit. A buffer was included so as to minimize the disturbances on the chaotic circuit due to the measurement apparatus. The diode used for the amplitude detection is a germanium diode in order to minimize the voltage drop. The frequency counter and the analog to digital converter (12 bits) are custom built PC-104 cards: this allows fast transfer time and precise control of the timings, allowing all successive periods to me measured during a given data acquisition time lapse.

An external circuit is used for detecting single periods and measuring their duration thanks to a comparator and a frequency counter with an internal clock at 14.31818 MHz. The amplitude is measured simultaneously by using a peak holder circuit and a 12 bit analog to digital converter. The trigger signal for the frequency measurement and the amplitude signal can be seen on Fig. 5. All consecutive periods can be measured for signals of frequency less than 1kHz.

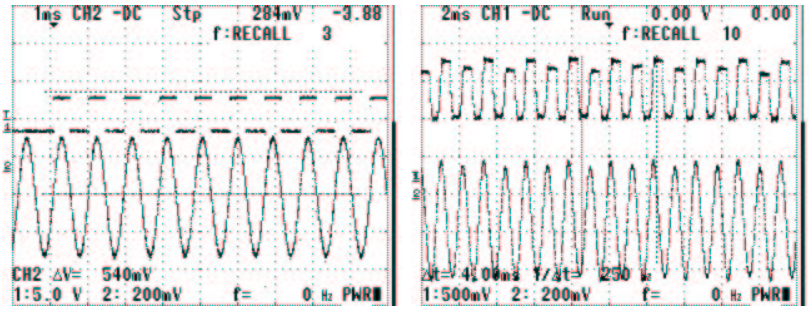

Fig. 5. Trigger signal for the frequency measurement (left, top curve) and amplitude measurement analog signal (right, top curve) output from the electronic circuit connected to Chua's circuit. On both images, the bottom curve is the signal observed at the output of Chua's circuit (chaotic regime). The amplitude signal (right) is read by a 12 bit analog to digital converter while a custom made frequency counter samples the pulse (left). Both values are read in less than 500 μs, half of the duration of a period of Chua's circuit, and allows the measurement of all successive periods.

4 Statistical analysis and bifurcation diagrams

All our statistical analysis, except the plotting of the bifurcation diagrams that were extended to high values of L on purpose, have been limited to regions of the L parameter for which the signal is observed to be periodic or pseudo-periodic with a constant mean value (as opposed to smaller oscillations overlapping large scale fluctuations such as the signals observed in the double scroll oscillator). This limitation is required by our data acquisition electronic circuit as well as by our intuitive definition of frequency and amplitude of pseudo-periodic signals.

We plotted the values taken by the amplitude and frequency time series versus the value of the L parameter in Fig. 6 and 7. One can observe the first bifurcation, followed by the chaotic behavior. Although these curves might appear quite noisy compared to the theoretical (Feigenbaum) diagram, the points far from the main path are not statistically significant, as can be seen by plotting the histogram distribution of the values. They are due to the fact that when recording a large number of consecutive points, one cannot avoid reading values to which outside electromagnetic noise is added. A statistical measurement of the instantaneous frequency and amplitude data (*i.e.* reading instantaneous values on randomly chosen periods) displays much cleaner diagrams (as the external noise source is statistically insignificant).

We also display the Allan deviation $\sigma_y(\tau)$: a -0.5 slope in the Allan deviation indicates a white frequency noise, while a -1 slope indicates $\alpha = +1$ (flicker phase fluctuations) or $\alpha = +2$ (white phase noise) frequency fluctuation spectra (Fig. 7).

We then display the evolution of various parameters which are characteristic of the dynamic of Chua's circuit for several chosen values of the L parameter: the frequency return maps (Fig. 8), the transfer functions (Fig. 9) and the amplitude

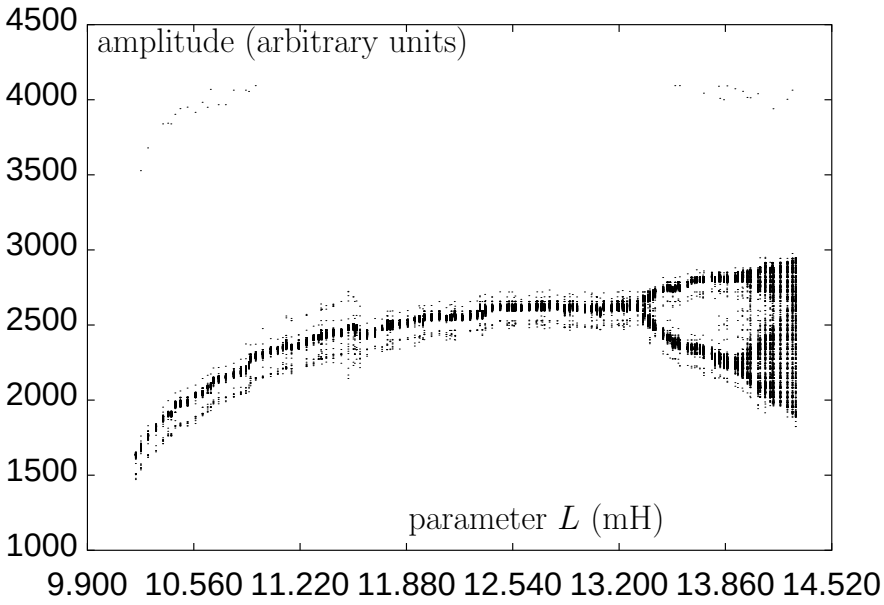

Fig. 6. Bifurcation diagram for the amplitude data.

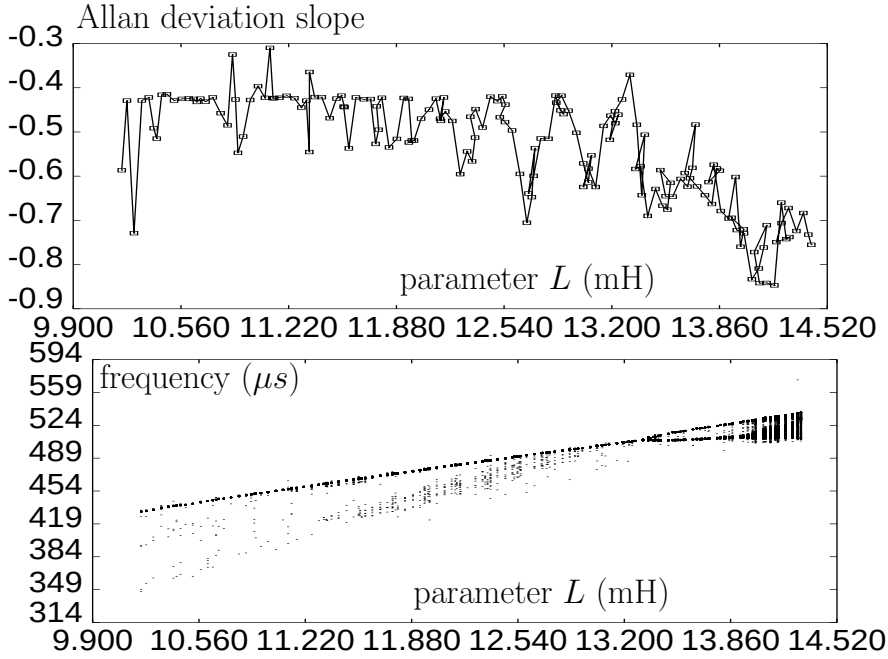

Fig. 7. Slope of Allan deviation (top) and bifurcation diagram (bottom) for the frequency data.

return maps (Fig. 10). We have selected values of the parameter for which we can see a periodic signal followed by signals characteristic of each further bifurcation, until obtaining a fully developed (Rössler like) chaotic attractor. The chosen values of L are the following: 13.167 mH for a periodic signal, 13.497 mH and 13.926 mH for signals after the first bifurcation, 14.207 mH for a signal after the second bifurcation, and finally 14.058 mH and 14.322 mH for fully developed chaos (Rössler-like attractor).

Figure 9 displays the evolution of the transfer functions with the parameter L. Absicissa are here given in duration units, rather than in the usual frequency units as classically used. The splitting of the fundamental spot (top-left) is clearly identified. A major difference between the return maps (defined by displaying f_{n+1} versus f_n when given a time series $(f_{n,n\in[1..N]})$) of the frequency data (Fig. 8) and the return maps of the amplitude data (Fig. 10) can be observed: the return maps of the frequency data are always an application (f_n defines one f_{n+1} only) while return maps of the amplitude data are no longer an application for large enough values of the parameter L. By adding additional dimensions, *i.e.* plotting a_n as a function of a_{n+1} and a_{n+2} $(a_{n,n\in[1..N]})$ being the amplitude data), the relation becomes an application again by unwrapping the return map, and the prediction is again possible (Fig. 11). We will see later the consequences in the prediction of these time series.

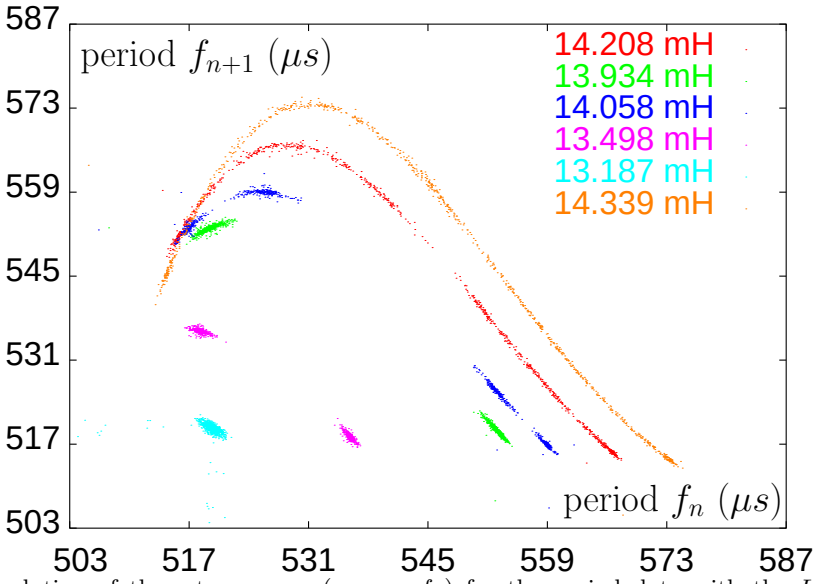

Fig. 8. Evolution of the return maps (a_n v.s. f_n) for the period data with the L parameter.

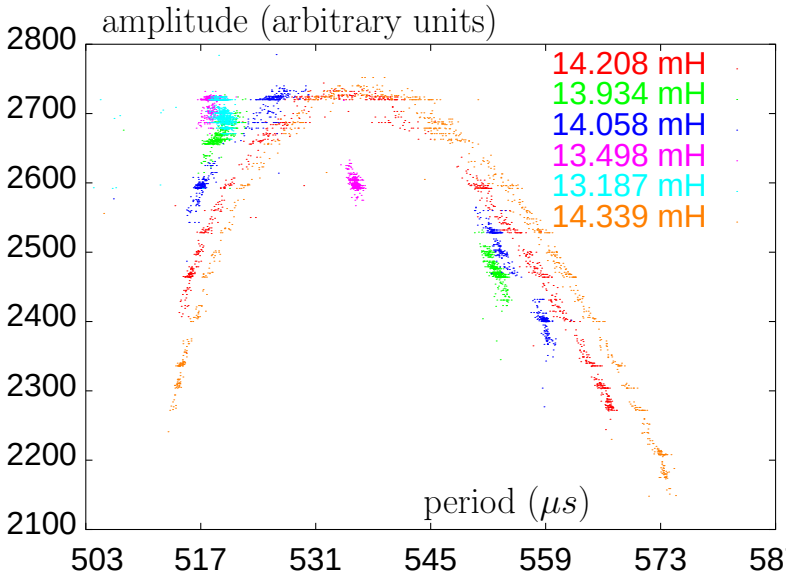

Fig. 9. Evolution of the transfer functions (the abscissa are not here the frequency as usually used when displaying transfer functions, but the period, which is more relevant in this study) with the parameter L.

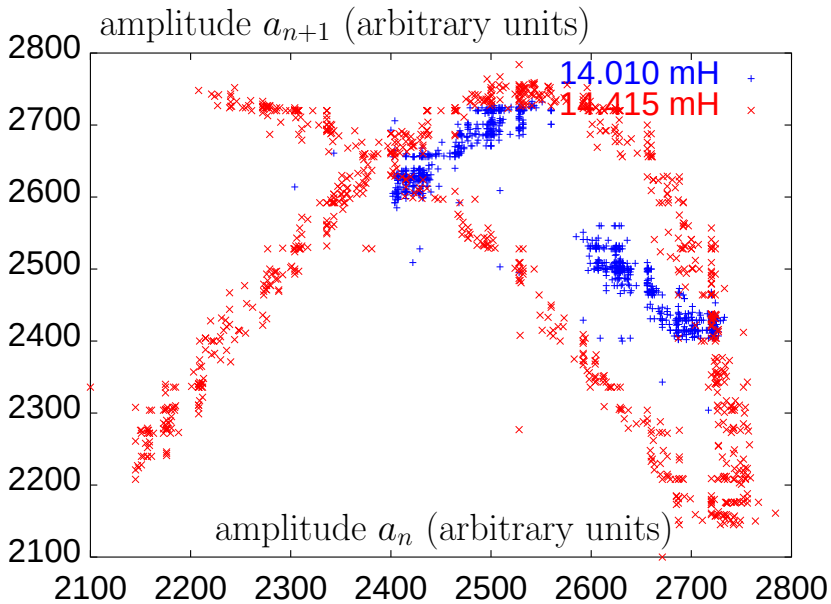

Fig. 10. Evolution of the return maps for the amplitude data with the L parameter. Notice that the relation between a_n and a_{n+1} (a_n being the amplitude data) is no longer an application for a value large enough of L.

5 Learning algorithms

Learning algorithms are aimed at computing a function f, learnt from examples; *i.e.* being given $(x_i, y_i)_{i\in\mathcal{L}}$ for $\mathcal{L}$ a given training set, a neural network looks for a function f such that foreach $i \in \mathcal{L}$, $f(x_i)$ is close to y_i.

Being given a time series $(X_i)_{i\in\mathbb{N}}$, we will try to make the neural network predict the behavior of the next elements after learning on a small sample. p-steps prediction is the prediction of (X_{n+p}), being only given $(X_i)_{i\leq n}$. The case $p = 1$ is the one-step prediction, the general case is p-steps prediction. Concretely, two approaches are possible:

• **Direct** methods: $x_i = (X_{i-m+1}, X_{i-m+2}, ..., X_i)$, and $y_i = X_{i+p}$. The neural network (or any other regression algorithm) finds f such that $f(x_i)$ is close to y_i. The obvious algorithm for predicting X_{n+p} is computing $f(X_{n-m+1}, X_{n-m+2}, ..., X_n)$.

• **Iterated** methods: $x_i = (X_{i-m+1}, X_{i-m+2}, ..., X_i)$, and $y_i = X_{i+1}$. The learning algorithm finds f such that $f(x_i)$ is close to y_i. Then we compute $\hat{y}_{n+1} = f(y_{n-m+1}, y_{n-m+2}, ..., y_n)$, $\hat{y}_{n+2} = f(y_{n-m+2}, y_{n-m+3}, ..., y_n, \hat{y}_{n+1})$, $\hat{y}_{n+3} = f(y_{n-m+3}, ..., y_n, \hat{y}_{n+1}, \hat{y}_{n+2})$, and so on until a prediction $\hat{y}_{n+p}$ of X_{n+p}.

In our experiments, we used the following algorithms, aimed at finding such a function f:

Backpropagation

One can find a description of this famous classical algorithm in [5]. We use it with or without weight decay, with various learning steps. The function f is parameterized by some weights, and these weights are chosen by a *gradient descent* on the quadratic error (called L_2 error), $\sum_{i\in\mathcal{L}}(y_i - f(x_i))^2$. The weight decay is aimed at regularizing f; it consists in replacing the L_2 error by $\sum_{i\in\mathcal{L}}(y_i - f(x_i))^2 + M\sum w_i^2$ (called the cost function).

Radial Basis Functions (RBF)

One can find a description of this algorithm in [5]. One places points $(C_j)_{j\in J}$ in the input space (the space of the (X_i)'s), as representative of the X_i's as possible, by the transfer algorithm, which is a variant of the k-means algorithm, experimentally more efficient. One can find a description of this algorithm in [12]. Then we minimize the quadratic error (in our experiments, modified through a regularization based upon minimization of the curvature, as explained in [5]), among functions of the form $x \mapsto \sum_{j\in J} \lambda_j \exp(-\frac{(x-C_j)^2}{\sigma_j^2}) + b$, the σ_j's being chosen heuristically, and the λ_j by the minimization of the cost function.

Support Vector Machines (SVM)

Support Vector Machines are recent learning machines described for example in [13,10] and mathematically more fully justified than most of other learning algorithms. They are theoretically based on the VC-dimension theory, and have been

used very efficiently on lots of benchmarks, especially when datasets are small with regard to the dimensionality. Support Vector Machines are aimed to have a little difference between empirical error and generalization error. They maximize the "margin" and minimize a ε-insensitive cost function ($\sum_{i\in\mathcal{L}} |f(x_i)-y_i|_\varepsilon$, with $|x|_\varepsilon = max(0, |x| - \varepsilon)$). SVM solve a convex quadratic problem with linear constraints (by considering the lagrangian of the function to be minimized), so they don't have, as backpropagation, problems of local minima.

We used only the usual RBF or polynomial kernels. Tests about the use of SVM with trigonometric polynomials on chaotic time series can be found in [10].

k-nearest neighbours

This algorithm, the simplest of all learning algorithms, simply looks at the k nearest neighbours $(x_{i_1}, x_{i_2}, ..., x_{i_k})$ of a point x among the learning samples, and computes the mean of their targets $(y_{i_1}, y_{i_2}, ..., y_{i_k})$ to find y supposed to be associated with x. This algorithm, which gave good results for one-step prediction, has been forgotten in the following experiments (long time prediction) because of its poor efficiency in this case.

Our experimental data sets are nearly 1000 points-long, and we work with 700 points for learning and 300 for performance evaluation.

6 Computation of the embedding dimension of a chaotic time series

We call **embedding dimension** the arity which is to be experimentally the most suitable for 1-step prediction. This arity is supposed optimal for n-steps predictions too. The embedding dimension is expected to be the smallest integer higher than the **order** (dimension of the attractors, bounded by the number of degree-one differential equations defining the system), which is experimentally observed for some of our cases.

We evaluate the embedding dimension by the following algorithm:

• Evaluation of the proportion of points predicted within 10%, 1%, 0.1%, 0.01% of the standard deviation, with any arity between 1 and 9. The algorithm chosen is the best among all the above cited one.

• Computing, for $X = 10\%, 1\%, 0.1\%, 0.01\%, 0.001\%$, which arity i_X is the most efficient, and with which the reduced deviation Z_X is significant.

• Computing, for each arity j, $Perf_j = \sum_{i_X=j} Z_X$

• Consider that the embedding dimension is j which maximizes $Perf_j$, provided that $Perf_j > Perf_k + 3$ for any $k \neq j$.

We show in table 1 the results in the case of a simulated dataset for a parameter value corresponding to a Rössler-like attractor (chaotic regime of the dynamical system).

We notice that the best results were obtained with SVM or RBF networks. The good behavior of SVM networks can be verified for long term prediction, whereas RBF will not be so efficient in this more difficult task.

Arity	0.001 %	0.01 %	0.1 %	1 %	10 %
1	0	0.73	7.64	31.27	62.55
2	0	3.64	30.91	93.45	100
3	0.36	4.01	40.15	99.27	100
4	0.36	2.19	22.99	92.70	100
5	0.36	0.73	5.47	60.58	100
6	0	0.73	4.40	35.16	99.27
7	0	0.73	3.30	27.11	90.48
8	0	0.37	1.83	22.34	90.11
9	0	0.37	1.83	12.45	89.74

Table 1. The number in column $x\%$ and row i is the smallest percentage of points which are predicted with error $< x\%$ of the standard deviation with an algorithm with arity i. The decrease of the generalization performance after the required arity is due to *overfitting* (*i.e.* there are too many parameters to be chosen for the finite number of points). Experimental acquisition noise can be reduced by increasing the number of inputs, so that the embedding dimension can become larger with regard to the order as noise increases. Here the chosen embedding dimension is three.

The results we obtained on our experimental datasets are summed up in table 2.

L (mH)	Evaluation of the order (amplitude)	$inf_{k\neq j} Z_j - Z_k$ for $j = argmax\ Z_j$ (amplitude)	Evaluation of the order (frequency)	$inf_{k\neq j} Z_j - Z_k$ for $j = argmax\ Z_j$ (frequency)
13.722	1*	4.0	1	16.0
13.868	1*	4.7	1	9.1
14.010	1	4.3	1	3.9
14.148	(3/4)		1	8.4
14.283	3	4.2	1*	7.5
14.415	4	8.6	1	10.7
simulation	3	37.5	N.A.	N.A.

Table 2. Estimated embedding dimension of the attractor. Only significant results are reported. Other ones were nearly following the same tendency; optimal arity is always equal to 1 for frequencies. For the amplitude data, 1 is optimal for $L \in [13.722; 14.010]$ mH, 3 for $L = 14.148$ mH (although results were not significant for this parameter value: arities of 3 or 4 were better than other arities, suggesting an embedding dimension of 3 or 4) and $L = 14.283$ mH, and 4 for $L = 14.415$ mH. For embedding dimensions marked with *, the results have been obtained while considering that the best arity is the last one which is significantly better than the previous ones (this has been done as the reduced deviation was not significant between the optimal arity and the following ones, as the overfitting was not strong enough). The third and fith columns numerically evaluate the statistical significance of the results. Simulated data logically lead to a much higher significance.

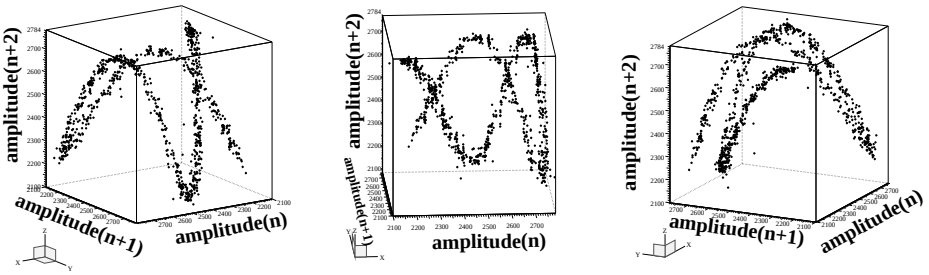

Fig. 11. 3D plots of the experimental amplitude return map at 1 and 2 steps in the chaotic regime of the circuit. $L = 14.415$ mH. These plotsw can be compared with fig. 10: adding a third dimension unwraps the return map.

7 Multi-steps prediction of chaotic time series

Fig. 12 shows an example of prediction of a time series using a neural network after learning the behavior of an experimental chaotic time series. A good agreement between experimental data and predicted data is obtained for 7 or 8 values in the future.

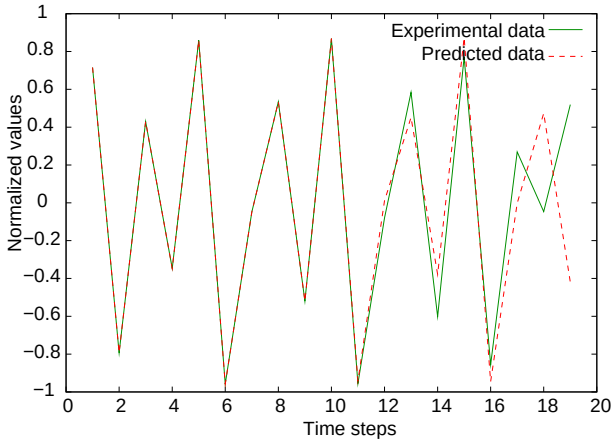

Fig. 12. Comparison between the experimental frequency data and predicted data after the learning phase is completed (the test set data for checking the ability of the neural network to predict the time series is always outside of the learning set). The 10 first points are experimental, and the 11^{th} point is the first one for which we compare the predicted and the real value. $L = 14.415\ mH$

Nb steps	Algorithm	L_1 error	L_2 error	10 %	1 %	0.1 %
2	Iterated RBF	$4.9\ 10^{-3}$	$6.30\ 10^{-5}$	1.00	0.86	0.23
4	Direct RBF	0.49	0.46	0.16	0.018	$3.6\ 10^{-3}$

Table 3. Comparison of the different algorithms (among the SVM, RBF and Backprop algorithms) for predicting a noise-free simulated time series of Chua's circuit for a parameter value leading to chaotic (Rössler like) behavior. The best results are obtained by the direct and iterated versions of the RBF algorithm. Neither algorithm was able to predict the time series for more than 8 steps in the future; the used arity is 3.

Number of steps	Algorithm	L_1 error	L_2 error	10 %	1 %	0.1 %	0.01 %
2	Iterated SVM (1)	0.070	0.0093	0.76	0.16	0.022	0.0036
4	Direct SVM (1)	0.20	0.085	0.33	0.040	0	0
8	Direct SVM (1)	0.32	0.20	0.24	0.023	0.0033	0
16	Direct SVM (2)	0.38	0.25	0.18	0.023	0.00000	0
32	Direct SVM (1)	0.51	0.43	0.18	0.047	0.031	0

Table 4. Description of the algorithm (among the SVM, RBF and Backprop algorithms) that gave the best prediction result for a given number of steps ahead. The data are the amplitude for a parameter value $L = 14.148$ mH. We used arity=3.

Number of steps	Algorithm	L_1 error	L_2 error	10 %	1 %	0.1 %
2	Iterated SVM(1)	0.12	0.028	0.56	0.076	0.0066
4	Direct SVM(1)	0.12	0.024	0.53	0.086	0.0099
8	Direct SVM(1)	0.13	0.033	0.55	0.056	0.0033
16	Direct SVM(1)	0.13	0.024	0.54	0.068	0.0032
32	Direct SVM(1)	0.13	0.028	0.52	0.093	0.012

Table 5. Same as table 4, for data which are the amplitude for a parameter value of $L = 13.722$ mH. We used arity=3.

In tables 3, 4 and 5, the algorithm cited at line i (second column) is the one which gave the best results for i steps prediction. The five numbers evaluating the performance are respectively the mean deviation divided by the standard deviation, the mean squared error divided by the variance, and the frequency of points predicted within 10%, 1% and 0.1% of the standard deviation. Table 3 sums up our results on our simulated dataset (Rössler-like attractor), table 5 reports results of experiments for a parameter value of $L = 13.722$ mH, and table 4 reports results of experiments for a parameter value of $L = 14.148$ mH.

Both RBF have no regularization and 100 centers.

SVM (1) is a degree 3 polynomial SVM, with $C = 1000/n$ with n the number of learning samples. SVM (2) is a degree 2 polynomial SVM, with $C = 10/n$.

We try to predict as efficiently as possible the most difficult data we had, *i.e.* the highest parameter value which is $L = 14.415$ mH for which the signal is

	$L = 14.148$, amplitude		
Number of steps	Backprop (tanh)	SVM (poly)	RBF
2	0.36	1.33	0.45
4	0.45	0.27	0.72
8	0.37	0.46	0.80
16	0.30	0.54	0.72
32	0.50	0.76	0.82

Table 6. Ratio of mean error for direct and iterated methods (low values mean a bad resistance to iteration). For each algorithm, the best results are taken into account.

still quasi-periodic, in the case of the frequency. The curves of Fig. 13 show the return map, iterated once, twice, ..., 8 times. We understand on this curve why the embedding dimension is 1 (it is clear that X_{n+1} is function of (X_n)), which makes possible the evaluation of the prediction horizon by viewing the iterates of the return map. On the other hand, the return map of the amplitude is not an application (Fig. 10) and an additional dimension must be added in order to unwrap the curve (Fig. 11) and define a_{n+2} as a function of a_{n+1} and a_n, $a_{n,n\in[1..N]}$ being the amplitude data.

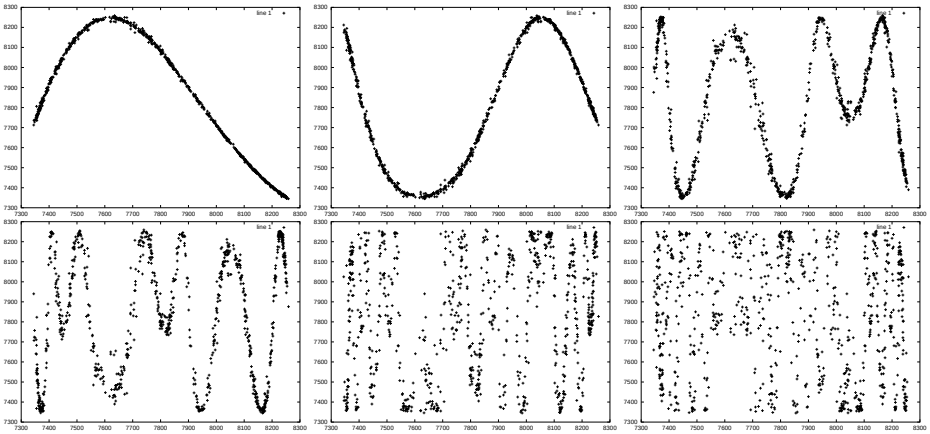

Fig. 13. Return maps of the frequency data f_n for (from left to right and top to bottom) 1, 2, 4, 5, 7 and 8 steps. $L = 14.415$ mH.

We can guess from these results that any prediction after 8 steps will be quite difficult; this is illustrated on the graph 14, showing the evolution of the L_2 depending upon the number of steps of the prediction. We can notice that as the optimal arity is 1, perhaps specialized methods for dimension 1 to dimension 1-regression could give better results; it is likely it would be difficult, even with much more precise regressions, to get a good prediction over many more steps.

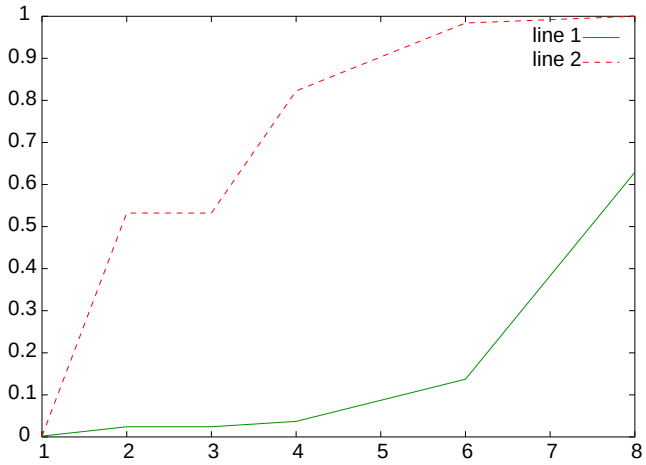

Fig. 14. Line 1 is the error observed with iterated SVM, and line 2 with direct SVM. The best result is obtained with the iterated method. That is only true for an arity of 1 or very short term predictions.

8 Conclusion

The results reported in table 6 compare iterated algorithms to direct ones, in different cases. The first point emphasized by these results (and verified in others time series too, simulated or experimental data) is the superiority of direct versions algorithms over iterated ones. Two exceptions are the short term prediction, and the case of embedding dimension 1, for which iterated algorithms have been significantly better. For 2 steps prediction, the best results have always been obtained with iterated SVM for experimental data or iterated RBF (simulated data, case without noise), sometimes very significantly.

With embedding dimension 1, Fig. 13 shows the superiority of iterated algorithms in the case of the frequency data for $L = 14.415$ mH. This has been verified on other files of embedding dimension 1, too. This can be intuitivly justified by the fact that iterated algorithms for p-steps prediction with embedding dimension k based upon a 1-step predictor f, approximates X_{n+p} by $f(\hat{X}_{n+p-k}, \hat{X}_{n+p-k+1}, ..., \hat{X}_{n+p-1})$, that, after replacement of $\hat{X}_i$ by formulas in $\{f, X_{n-k+1}, X_{n-k+2}, ..., X_n\}$, involves a number of f's exponential in p if $k \geq 2$ (increasing as Fibonacci numbers if $k = 2$), and linear if $k = 1$.

The second point is that SVM (which here compute polynomials) or RBF are much more resistant to iterations than backpropagation with tanh. We can try to justify this result by the following remarks:

- [11] predicts a bad behavior of iterated neural networks with tanh
- The argument used in [11] about tanh cannot be generalized to polynomials or gaussians; polynomials and gaussians can easily generate cycles of any lengths (see [6], [7], [9]), without increasing the degree/number of gaussians. [11] predicts a bad asymptotic behavior of iterated neural networks with tanh activation

functions, as the long term prediction depends upon attractor dynamics closely related to the architecture. This explains why iterated methods (using tanh) do not resist to long term predictions and is confirmed by our experimental results. Methods using gaussians or polynomials seem to be much more iteration-resistant, but anyway for long term prediction, direct algorithms seem to be more efficient (provided that the embedding dimension is bigger than 1 as explained above).

The problem of cost functions has been emphasized [8]. It seems that ε-insensitive cost functions are more adapted to long time prediction than quadratic cost functions. This can be explained intuitively by studying two different approaches to the problem of long term prediction of chaotic time series:

1) Try to predict exactly where the point will be. a) Any error will be punished. b) Big errors are much more important than small errors.

2) Try to be close to the point. a) Small errors are not important. b) Big errors are not much more important than medium errors.

The first "punishment algorithm" is unjust because:

• A small error can be due to measurement errors. So point a) is more pertinent in case 2).

• A big error is equivalent to a medium error: if the error is not small, it means that you have not followed the good trajectory. Once the trajectory is missed, the distance from it is not important, so b) is more pertinent in case two.

The prediction horizon does not seem to be very disturbed by noise but rather by the chaotic nature of the dataset. Whereas experimental data are never predicted with a very good precision (which is an intuitive result as a minimum error due to noise is always present), they can be predicted more efficiently for long term prediction than the simulated data (for a parameter in the chaotic regime) as the sensitivity to initial conditions has more influence in this latter case.

Our objective of predicting the frequency fluctuations, whether from deterministic signals or non-white noise, is based on the hypothesis that they are of deterministic origin (*i.e.* with non-null autocorrelation).

We have thus used neural networks to predict properties and the evolution of a non-linear deterministic oscillator (Chua's circuit) without explicitly informing them of the underlying physics. We can extrapolate these results to quartz oscillators after considering their topological similarities. However, although we have been able to successfully predict the fluctuations of an oscillator, these informations must be completed by an evaluation of the parameter value if they are to be used for correcting the frequency noise: acting on the physical system (by changing the value of a capacitor connected to the quartz resonator for example – point C in Fig. 2, left) changes the physical properties of the device the neural network has been trained on. By adding the parameter value as a variable of the neural network, we include any change on the physical system aimed at correcting the frequency fluctuations. The learning phase is however

more difficult as a whole set of data (for various values of the parameter) must be learnt.

References

1. Leon O. Chua and Gui-Nian Lin, *Canonical Realization of Chua's circuit family*, IEEE Transaction on Circuits and Systems **37** (7) (1990)
2. G. Sarafian and B. Z. Kaplan, *Is the Colpitts a relative of Chua's circuit?*, IEEE Transaction on Circuits and Systems **42** (6) (1995)
3. Michael Peter Kennedy, *On the relationship between the chaotic Colpitts oscillator and Chua's oscillator*, IEEE Transaction on Circuits and Systems **42** (6) (1995)
4. Thomas P. Weldon, *An inductorless double scroll chaotic circuit*, Am. J. Phys **58** (10), October 1990
5. C.M. Bishop, *Neural Networks for Pattern Recognition*, Oxford (1995)
6. M. Feigenbaum, *Quantitative universality for a class of nonlinear transformations*, J. Statist. Physics 19 (1978), pp 158-185
7. M. Feigenbaum, *The universal metric properties of nonlinear transformations*, J. Statist. Physics 21 (1979), pp 669-706
8. S. Haykin, J.Principle, *Using Neural Networks to Dynamically model Chaotic events such as sea clutter; making sense of a complex world*, IEEE Signal Processing Magazine 66:81 (1998)
9. O. Lanford, *A computer-assisted proof of the Feigenbaum conjectures*, Bull. American Math. Soc. 6 (1982), p. 427-434
10. S. Mukherjee, E. Osuna, F. Girosi, *Non-linear Prediction of Chaotic Time Series Using Support Vector Machines*, Proc. of IEEE NNSP'97, Amelia Island, FL (1997)
11. A. Priel, I. Kanter, D.-A. Kessler, *Analytical study of the interplay between architecture and predictability*, NIPS'10, proceedings of the 1997's conference, MIT Press (1998), pp 315-321
12. S. Thiria, Y. Lechevallier, O. Gascuel, S. Canu, *Statistique et méthodes neuronales*, Dunod (1997)
13. V.N. Vapnik, *The Nature of Statistical Learning*, Springer (1995)
14. C. Eckert, J. Miehe and M. Planat, *On the hidden order in the frequency noise of an electronic oscillator*, Physical Review E **54** (1996), p. 6093
15. P. Horowitz and W. Hill, *The art of electronics*, Cambridge University Press (1989)