A GENETIC ALGORITHM BASED DESIGN APPROACH FOR THE PROPERTIES OF A GAUSSIAN PROCESS FOR TIME SERIES FORECASTING

J. Baumgartner, C. Rodríguez Rivero and J. Pucheta

Departments of Electrical and Electronic Engineering, Laboratory of Research in Applied Mathematics Control (LIMAC), at Faculty of Exact, Physical and Natural Sciences - National University of Córdoba, Córdoba, Argentina. josef.s.baumgartner@gmail.com, cristian.rodriguezrivero@gmail.com, julian.pucheta@gmail.com

Abstract: In this work autoregressive filters based on Gaussian processes are used to forecast time series. As a benchmark function the Mackey-Glass delay differential equation is used. The time lagged inputs of the filters and their covariance functions are determined via a multicriteria, genetic algorithm – namely the NSGA-II. The used optimization criteria are the number of inputs and the prediction error of the filters on the known data which leads to Pareto optimal solutions. The obtained filters are validated with out-of-sample data and their properties are discussed in the context of the given problem.

Keywords: Time series forecasting, Mackey-Glass, NSGA-II, Gaussian processes.

1. INTRODUCTION

Natural phenomena prediction is a challenging topic, useful for control problems from agricultural activities. Before starting with the agriculture venture, the availability of estimated scenarios for water predictability would help the producer to decide. There are several approaches based on neural networks (NN) that face the rainfall forecast problem for energy demand purposes (Chow *et al.*, 1996), for water availability (Liu *et al.*, 1999) and for seeding growth (Patiño *et al.*, 2007). Thereby the forecasted time series is either taken from an ensemble of measurement points or from a benchmark function.

In this work the Mackey-Glass equation is used as a benchmark function to test a new approach to time series forecasting based on Gaussian processes (GP) which belong to the class of kernel methods.

1.1. Overview of the Mackey-Glass equation

The Mackey-Glass equation (MG) serves to model natural phenomena. It has been used by various authors to compare different techniques for foretelling and regression models (Velásquez *et al.*, 2004). Therefore the MG equation is chosen as a benchmark function in this work to validate the proposed approach to time series forecasting.

The MG equation is explained by the time delay differential equation defined as

$$\dot{y}(t) = \frac{\alpha y(t-\tau)}{1+y^{c}(t-\tau)} - \beta y(t) \qquad (1)$$

where *α*, *β*, and *c* are parameters and *τ* is the delay time (Glass *et al.*, 1988). According as *τ* increases, the solution turns from periodic to chaotic. Equation (1) is solved by a standard fourth order Runge-Kutta integration step, and the series to forecast is formed by sampling values with a given time interval.

Thus, a time series with a random-like behavior is obtained, and the long-term behavior changes thoroughly by changing the initial conditions. Furthermore, by setting the parameter β ranging between 0.3 and 0.8 the stochastic dependence of the deterministic time series obtained varies according to its roughness.

1.2. The proposed algorithm

One of the motivations for this study follows the closed-loop control scheme where the controller considers meteorological future conditions for designing the control law as shown in Fig. 1. In that scheme the controller considers the actual state of the crop by a state observer and the meteorological variables, referred by $x_{(k)}$ and R_o, respectively. However, this work is only dedicated to the prediction part of the controller.

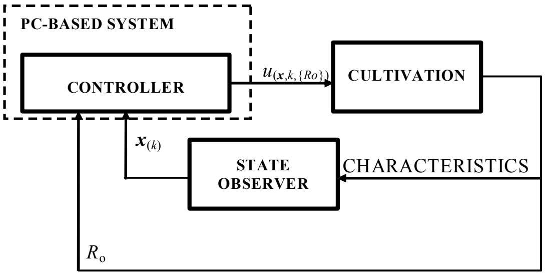

Fig. 1. Closed-loop PC-based control approach in which a sequence of future meteorological conditions is considered.

The predictions for the controller are based on the past values of a variable which can be interpreted as a time series. Using this time series, an autoregressive filter can be implemented as a one-step-ahead predictor for the observed values. This procedure was applied successfully in many publications (Pucheta *et al.*, 2007a; Velásquez *et al.*, 2004), where the filters were realized as NN.

The new aspect of this work is the application of a nonparametric, autoregressive filter whose properties are optimized by a multicriteria, genetic algorithm. The advantage of nonparametric filters over ARMA NN-based filters is that very little a priori knowledge of the modeled data is needed because there are no parameters – like e.g. the number of layers of a NN – that have to be defined a priori.

2. PROBLEM FORMULATION

This work deals with a classical prediction problem. Given the past states of a process one is interested in forecasting some future states of this process. Thereby the states are equally distributed over time, i.e. $x(t\text{-}T)$, $x(t\text{-}2T)$, . . . , $x(t\text{-}mT)$, where T is the sampling period and m is the prediction order.

The main issue when forecasting a time series is how to retrieve the maximum possible information from the known data. In other words one is interested in adjusting the parameters of a filter such that the given data is explained by the filter in the most accurate way.

Most approaches to time series forecasting deal with parametric filters like NN whose coefficients are adapted according to a learning rule (Pucheta *et al.*, 2009). In this work however, nonparametric filters – namely GP – with covariance functions are used. Thus, one of the main problems is to find a covariance function that works well with the given data.

Besides the covariance function, the time lags of the inputs of the autoregressive filter need to be defined according to the long and short term dependences of the given time series. Thereby one has to keep in mind that the covariance function and the time lagged inputs are interfering with each other. As a result both optimization problems need to be solved simultaneously.

Hence, one goal of this work is to show that the applied optimization algorithm converges for the given problem. Apart from that it has to be shown that the found combinations of a covariance function and a set of inputs lead to a GP filter that predicts the given time series with a small error.

The described approach is evaluated with a time series that is taken from the MG equation (1). As training data we use the first 102 values whereas the following 18 values are used as validation data.

3. PROPOSED APPROACH

This work describes an approach to select the inputs and the covariance function of a nonparametric filter. Both problems – finding a covariance function and determining the inputs of the filter model – are discrete, because the covariance function is chosen out of a discrete set of possible functions and the inputs are defined by discrete time lags. To handle these two optimization problems a genetic, nondominated sorting algorithm called NSGA-II (Deb, 2002 *et al.*) is used.

3.1. APPLICATION OF THE NSGA-II

The NSGA-II is a multiobjective genetic algorithm that showed good results for various optimization problems (Correa *et al.*, 2008; Deb, 2002 *et al.*). The algorithm starts with random individuals which represent the first generation. After the evaluation of the first generation the best individuals are chosen as parents and the next generation is created via mutation and crossover operations of the parents. Repeating this procedure for various generations leads to optimized individuals.

In this work each individual represents a GP with a covariance function and certain time lags. Via mutation and crossover operators the covariance function and the time lags can change independently from generation to generation. Hence the NSGA-II is searching for an optimal combination of a covariance function and time lagged inputs.

To apply the NSGA-II one has to define the optimization criteria – called fitness values – which serve to classify the individuals of one generation. Thereby two goals should be kept in mind. On the one hand the filter model has to be as accurate as possible. On the other hand the number of inputs of the Gaussian Process should be as small as possible to avoid overfitting.

The first fitness value is a measure for the accuracy of the GP filter. To evaluate this fitness value the prediction error on out-of-sample data is calculated. Therefore the given 102 values are split into two parts so that the filter can be trained with the first 82 values before it is evaluated on the remaining 20 values. Keeping in mind that the time lags of the GP are limited to 30, the first training point has the index 31. Otherwise it is impossible to create the input vector of the GP. Hence, 52 training points can be created out of the 82 data points.

To compare the performance of different filters the Symmetric Mean Absolute Percent Error (SMAPE) is calculated for each filter. The SMAPE is defined as

$$SMAPE = \frac{1}{n}\sum_{t=1}^{n} \frac{|X_t - F_t|}{(X_t + F_t)/2}. \qquad (2)$$

Here t is the observation time, n is the size of the test set, X_t and F_t are the actual and the forecasted time series values at time t respectively. The SMAPE calculates the symmetric absolute error between the actual X_t and its corresponding forecast value F_t for the specified part of the given data. Thus one of the optimization criteria is to minimize the SMAPE of the filter model by finding an optimal subset of inputs together with a covariance function.

The second fitness value is the number of inputs of the filter, i.e. the number of past values that are used to predict the next value. To avoid a high input dimension, which might lead to overfitting, the filters with fewer inputs are preferred by the genetic algorithm.

The two defined fitness values are competing because a filter with only few inputs does not have much information about the time series and will thus tend to have a higher SMAPE. Therefore the NSGA-II will not find one best solution but various solutions that represent different trade-offs between the SMAPE and the number of inputs. These trade-off solutions lie on the so called Pareto Front. For example one solution might be a filter with few inputs but a high SMAPE whereas a filter with more inputs leads to a lower SMAPE.

After running the NSGA-II one has to choose the individual from the Pareto Front that marks the desired trade off between the two fitness values. In this case the focus is set on the filter with the smallest SMAPE which means that at the time when the best individual is chosen no attention is paid to the number of inputs.

To validate the results of the described optimization algorithm, more than one *multistart* can be executed. This means that the algorithm is started several times from a random population.

3.2. Parameters of the NSGA-II

There are several parameters that have to be defined when running the NSGA-II. Besides the inner parameters of the genetic algorithm there are various problem-specific parameters that have influence on the found solutions.

In this work the population of the NSGA-II consists of 200 individuals that are evaluated over 200 generations. Hence the fitness values of 40.000 individuals are calculated in each multistart. After the evaluation of one generation new individuals are created via mutation with a probability of 10% and via crossover operations with a probability of 90%.

Each individual must have between 1 and 20 inputs from the past and none of these inputs can have a time lag less than 1 or greater than 30. In other words $1 \le i \le 20$, $min(m_1,...,m_i) \ge 1$ and $max(m_1,...,m_i) \le 30$ for

$$y^*(t) = GP(x(t - m_1 T),...,x(t - m_i T)) \qquad (3)$$

where $GP(\cdot)$ is the Gaussian Process, $y^*(t)$ is the predicted value at time t and T is the sampling period.

For this setup there are more than $6 \cdot 10^8$ possible combinations of inputs from the past. In Section 4 it is shown however, that the 40.000 evaluations of each multistart are sufficient to obtain good approximations of the Pareto Front.

The mentioned parameters describe the optimization process that searches for optimal sets of inputs in combination with a covariance function. Thereby the covariance function is taken from a set of possible functions where each function has special properties which might lead to accurate filters for certain inputs. The used covariance functions are:

- linear covariance function with ARD;
- squared exponential covariance function with ARD;
- neural network covariance function;
- isotropic rational quadratic covariance function;
- Matern covariance function with υ=3/2 and υ=5/ 2 (Matern, 1960);

- the sum of a linear covariance function with ARD and a white noise covariance function;
- the sum of a squared exponential covariance function with ARD and a white noise covariance function;
- the sum of a neural network covariance function and a white noise covariance function;
- the sum of a isotropic rational quadratic covariance function and a white noise covariance function.

Thereby ARD stands for automatic relevance determination (Neal, 1996). A detailed description of these covariance functions can be found in (Rasmussen *et al.*, 2006).

In total seven multistarts are carried out with the described configuration to validate the convergence properties of the NSGA-II in the given case.

3.3. Training and Prediction of the Gaussian Process

The training of the GP consists of two stages. First of all the training data is constructed out of the given time series according to the defined time lags.

Then the GP filter is tuned with the obtained training data by varying the so called hyperparameters of the covariance function. Depending on the covariance function there are several hyperparameters available that need to be adjusted to suit the training data. In other words one is interested in finding a maximum of the *log marginal likelihood*. Without going into detail the framework presented in (Rasmussen *et al.*, 2006) is used to optimize the hyperparameters in this work. Once the hyperparameters are found the training process is finished.

To evaluate a GP filter one has to calculate the covariance matrix K and its inverse K^{-1}. For n given training points K has size (nxn). Its entries are the pairwise covariances of the training inputs which makes K a symmetric matrix. Supposing that the variables have a joint Gaussian distribution with zero mean, the mean prediction for an unknown input f^* is given by

$$f^* = K(X^*, X)K(X, X)^{-1} f$$

where X^* is the unknown input, X are the training inputs and f are the training outputs. If the mean of the data is not zero, it can be transformed straightforward to fit the conditions.

4. MAIN RESULTS

In this section the proposed approach is tested for different parameter settings of the MG equation. Thereby special attention is paid to the results of the genetic algorithm because it is an important part of the described modeling process.

4.1. Time series from MG equations

The used time series are obtained from the MG equation (1) with different values for the parameter β. The variations of β and the resulting Hurst parameters H are shown in Table 1. For each β 120 data points were created from which the first 102 points are used to adjust the filter model while the last 18 points serve as validation data.

To characterize each time series, Hurst's parameter H is used. This parameter is an indicator for the roughness of a time series where a high value of H indicates a smooth series. Hence the first series with β = 0.3 is the smoothest whereas the third series with β = 0.8 is the roughest.

Table 1 Parameters of different time series obtained from the MG equation.

Series No.	β	H
1	0.3	0.92
2	0.5	0.58
3	0.8	0.46

4.2. Results of the NSGA-II

For each given time series seven multistarts of the NSGA-II were evaluated. Exemplary the Pareto Fronts of the multistarts of the time series with β = 0.8 are shown in Fig. 2. One can see that each multistart found almost the same Pareto Front. This is a strong indicator that the chosen population size and the number of generations were sufficient.

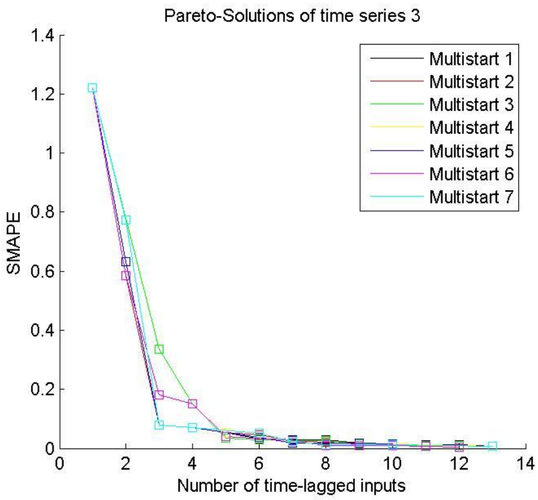

Fig. 2: Pareto Fronts of seven multistarts for the time series β = 0.8.

Once all multistarts are finished the best trade-off solution has to be chosen from the Pareto Front. While running the NSGA-II the number of time lagged inputs was an optimization criterion in order to reduce the dimensionality of the GP filter. Still the main aspect of this work is the prediction of a time series with the smallest possible SMAPE as defined by equation (2). For that reason the individual with the smallest SMAPE of all multistarts is chosen for the final model.

The GP is realized as described by equation (3) together with a covariance function.

For the time series defined by β = 0.3 the model with the smallest SMAPE has the coefficients m_i = {3,7,9,10,17,25,26} and a rational quadratic covariance function with noise.

The time series for β = 0.5 is best modeled by the time lagged inputs m_i = {3,6,7,8,9,11,12,15,17,20,23,24}. The corresponding covariance function is a matern function with υ=3/2.

In the case β = 0.8 the best combination of time lagged inputs is m_i = {3,4,5,7,9,10,12,13,16,17,23,25}. These inputs require a matern covariance function with υ=3/2.

Besides the best configurations of all multistarts the best covariance functions of each multistart are shown in Table 2. The results are discussed in the next section.

Table 2. Selected covariance functions of the NSGA-II. The covariance function of the best individual of each multistart is counted.

Series No.	Name of Covariance Function	Best covariance function in 7 multistarts
1	squared exponential	1
	squared exponential + white noise	2
	rational quadratic + white noise	4
2	neural network + white noise	2
	matern function, υ=3/2	2
	rational quadratic	3
3	squared exponential	3
	matern function, υ=3/2	4

4.3. *Prediction results for the MG time series*

The GP filters are trained with the first 102 values of a time series predictions are made for the following 18 points. In the training process these data points were not taken into account, hence they are out of sample data.

In Fig. 3 the predictions of the GP filters and the data of the time series are plotted. Considering the SMAPE of each prediction and the corresponding Hurst's parameter as shown in Table 1 it can be seen that a time series with a high Hurst's parameter H is easier to predict than one with a low value of H.

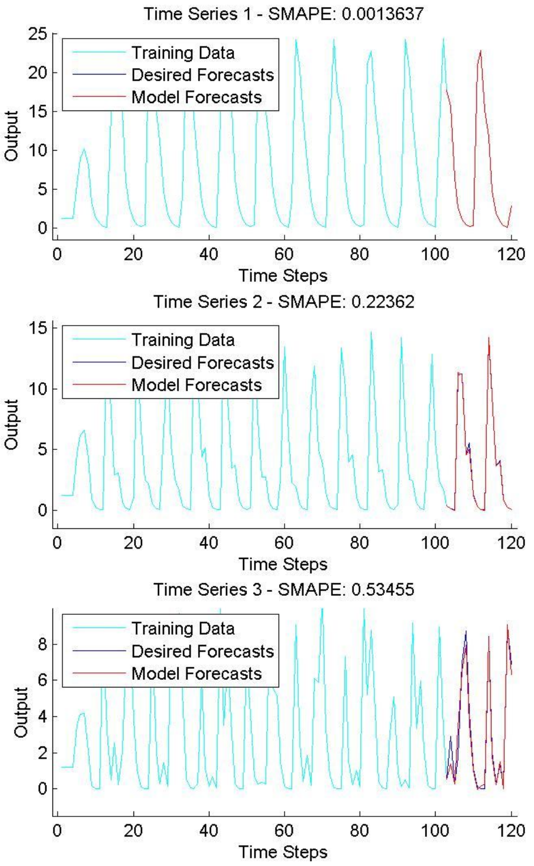

Fig. 3: Prediction results of the GP filters for the time series with β = 0.3, β = 0.5 and β = 0.8.

5. DISCUSSION

In this work an approach to time series forecasting was presented. Because of the long runtimes of the NSGA-II the described approach was only used to predict the last values of a time series and not to make online predictions.

Regarding the results of the NSGA-II and the predictions from the previous section one can see the influence of the smoothness of the time series. In this case the smoothness depends basically on the parameter β in the MG equation. For low values of β smooth time series are created which are best modeled by standard covariance functions like a squared exponential or a rational quadratic covariance function with noise.

For rougher time series the aggregation of noise to the covariance function becomes more redundant and the matern covariance function (Matern, 1960) with υ=3/2 gets more important as β increases.

The complexity of the time series is not only reflected by the selected covariance functions but also by the number of used time lags. In the case β = 0.3 only seven time lags are used whereas in the case β = 0.8 the GP has 14 time lagged inputs. This means also that the NSGA-II did not use the allowed maximum of 20 time lagged inputs.

Besides the number of time lagged inputs it has to be noticed that none of the time lags is greater than 26. Keeping in mind that the NSGA-II could have chosen time lags up to 30 this result is a confirmation that the NSGA-II was run with a sufficient degree of freedom concerning the time lags.

6. CONCLUSIONS

In this work, a genetic algorithm based design approach for the properties of a Gaussian process for time series forecasting was presented. The proposed approach is quite different to other filter approaches for time series forecasting. Instead of a NN based filter a nonparametric filter based on a GP is used. The parameters of the GP filter were adjusted offline by a genetic algorithm that requires several data points to work properly. For that reason only forecasts of the last 18 points of the time series were made.

Although the described algorithm is straightforward, the results obtained from a classical benchmark function like the MG equation are encouraging, especially for a smooth behavior of time series. Because of the special properties of this approach, it is difficult to compare the results with other papers. Still the approach deserves another study with real data.

BIBLIOGRAPHY

Chow, T. W. S. and Leung, C.T. (1996). Neural network based short-term load forecasting using weather compensation. *IEEE Transactions on Power Systems,* **Vol.11**, Iss.4, pp. 1736-1742.

Correa Florez, C. A., Bolaños, R. A. and Cabrera A. M. (2008). Algoritmo Multiobjectivo NSGA-II Aplicado Al Problema De La Mochila. In: *Scientia et Technica Año XIV,* **Vol.39**, Universidad Tecnológica de Pereira.

Deb K., Pratap A., Agarwal S. y Meyarivan T. (2002). A Fast and Elitist Multiobjective Genertic Algorithm. In: *IEEE Transactions on evolutionary computation*, **Vol.35**, No.2.

Glass L. and Mackey, M. C. (1988). From Clocks to Chaos, The Rhythms of Life. *Princeton University Press*, Princeton, NJ.

Liu, J. N. K. and Lee, R. S. T. (1999). Rainfall forecasting from multiple point sources using neural networks. In: *Proc. of the International Conference on Systems, Man, and Cybernetics.* **Vol.3**, pp. 429-434.

Matern, B. (1960). Spatial Variation. Meddelanden fran Statens Skogsforskningsinstitut, 49, No.5. Almaanna Forlaget, Stockholm. Second edition, Springer-Verlag, Berlin, pp. 85, 87, 89.

Neal, R. M. (1996). Bayesian Learning for Neural Networks. Springer, New York. Lectures in Statistics 118.

Patiño, H. D., Pucheta, J., Schugurensky, C., Fullana, R. and Kuchen, B. (2007). Approximate Optimal Control-Based Neurocontroller with a State Observation System for Seedlings Growth in Greenhouse. In: *IEEE International Symposium on Approximate Dynamic Programming and Reinforcement Learning.* pp.318-323.

Pucheta, J., Patiño, H. D. and Kuchen, B. (2007a). Neural Networks-Based Time Series Prediction Using Long and Short Term Dependence in the Learning Process. In: *Proc. of the 2007 International Symposium on Forecasting*, 24th to 27th of June 2007 Marriott Marquis Times Square, New York.

Pucheta, J., Patiño, H., Schugurensky, C., Fullana, R. and Kuchen, B. (2007b). Optimal Control Based-Neurocontroller to Guide the Crop Growth under Perturbations. *Dynamics Of Continuous, Discrete And Impulsive Systems Special Volume Advances in Neural Networks-Theory and Applications. DCDIS A Supplement, Advances in Neural Networks, Watam Press*, **Vol.14** (S1), pp. 618-623.

Pucheta, J., Patiño, D. and Kuchen, B. (2009). A Statistically Dependent Approach For The Monthly Rainfall Forecast from One Point Observations. In: *IFIP International Federation for Information Processing Volume 294, Computer and Computing Technologies in Agriculture* II, **Vol.2**, eds. D. Li, Z. Chunjiang, (Boston: Springer), pp. 787–798.

Rasmussen, C. E. and Williams C. K. I. (2006). Gaussian Processes for Machine Learning. The MIT Press.

Velásquez, J. D. (2004). Pronóstico de la serie de Mackey Glass usando modelos de regresión no-lineal. In: Dyna, *Revista De la Facultad de Minas – Universidad Nacional de Colombia – Sede Medellín.* **Vol.71,** No.142. pp. 85-95.