

# Chapter 31

## Evolutionary Identification of Block-Structured Systems

M. Schoukens and K. Worden

**Abstract** The identification of a nonlinear model often involves a significant amount of user interaction. The proposed SADE evolutionary algorithm-based identification approach for block-structured systems reduces this user interaction to a minimum. This is illustrated in this paper for the Wiener-Hammerstein class of systems. On top of this, most of the assumptions and limitations on the considered Wiener-Hammerstein system class can be omitted compared to the popular BLA and correlation based approaches. The developed identification algorithm is applied on the 2009 Wiener-Hammerstein benchmark to illustrate its good performance.

**Keywords** Wiener-Hammerstein • Block-Oriented • Nonlinear Systems • System Identification • Evolutionary Algorithms

### 31.1 Introduction

Nonlinear models are becoming more and more important nowadays to obtain a better insight in the behaviour of the system under test, to compensate for a potential nonlinear behaviour, or to improve control performance. One of the more popular nonlinear model structures is the block-oriented model [1]. Block-oriented nonlinear models are quite simple to understand and easy to use, due to the separation of the nonlinear dynamic behaviour into linear time invariant (LTI) dynamics and static nonlinearities. The Wiener-Hammerstein system class can be seen as a generalisation of the popular Wiener and Hammerstein system classes. A Wiener-Hammerstein system is a block oriented system where the static nonlinearity is sandwiched in between two LTI blocks (see Fig. 31.1).

The problem of identifying a Wiener-Hammerstein system is challenging since the nonlinear subsystem is inaccessible from both the input and the output. A variety of Wiener-Hammerstein identification methods have been developed over the last years using different approaches. Two nonparametric methods are described in [2, 3] using carefully designed input signals, Volterra (and tensor decomposition) based approaches are presented in [4–6], and some methods use evolutionary algorithms [7–9] to solve the nonlinear optimisation problem. A wide range of approaches use the Best Linear Approximation (BLA) [10, 11], or a similar correlation analysis, as a starting point for the algorithm, e.g. [3, 8, 12–16]. The interest in the Wiener-Hammerstein identification problem is also illustrated by the two Wiener-Hammerstein benchmarks that are available online [17, 18].

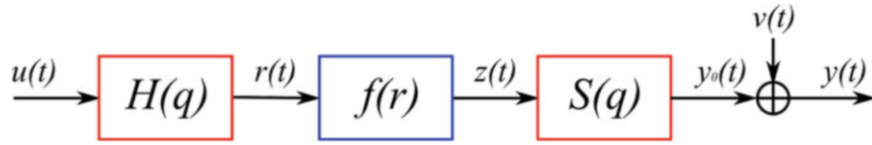
This paper proposes to use the SADE evolutionary algorithm optimisation approach [19, 20] to tackle the Wiener-Hammerstein identification problem in a user-friendly way. The SADE algorithm has proven its robustness already in the past on the identification of hysteretic systems [21, 22]. The proposed identification algorithm requires very little user interaction. On top of this, most of the assumptions and limitations on the system class can be omitted compared to the BLA and correlation based approaches.

The proposed evolutionary approach is very different from the one reported in [8], where the evolutionary optimisation is used only for the pole-zero allocation problem reported in [13]. The approaches that are presented in [7, 9] are more similar to the method presented in this paper. However, [7] only considers the problem where the LTI blocks are represented by a

---

M. Schoukens  
Department of ELEC, Vrije Universiteit Brussel, Pleinlaan 2, B-1050, Brussels, Belgium

K. Worden (✉)  
Dynamics Research Group, Department of Mechanical Engineering, University of Sheffield, Mappin Street, S1 3JD, Sheffield, UK  
e-mail: [k.worden@sheffield.ac.uk](mailto:k.worden@sheffield.ac.uk)



**Fig. 31.1** A Wiener-Hammerstein system consists of a static nonlinear block  $f(r)$  sandwiched in between two LTI blocks  $H(q)$  and  $S(q)$

FIR model, and a simplified differential evolution algorithm is used. The method presented in [9] uses a biosocial culture algorithm. It is comparable to the approach presented here, although it requires more hyperparameters to be selected by the user.

The remainder of the paper introduces the Wiener-Hammerstein identification problem (Sect. 31.2), discusses the evolutionary algorithm-based identification method (Sect. 31.3) and illustrates the effectiveness of the method in Sect. 31.4 on the 2009 Wiener-Hammerstein benchmark that was studied at the IFAC SYSID conference in 2009 [17].

## 31.2 Problem Formulation

Wiener-Hammerstein systems consist of a static nonlinear block that is sandwiched in between two LTI blocks (Fig. 31.1). The output  $y(t)$  of a Wiener-Hammerstein system is given by:

$$y(t) = S(q) [f(H(q) [u(t)])] + v(t), \quad (31.1)$$

where  $u(t)$  is the known input signal, and  $v(t)$  is an unknown additive disturbance with a finite variance acting on the output only. It is assumed that the input  $u(t)$  is persistently exciting the system under test. The system blocks  $H(q)$ ,  $S(q)$  and  $f(r)$  are given by:

$$H(q) = \frac{D(q)}{C(q)} = \frac{d_0 + d_1 q^{-1} + \dots + d_{n_d} q^{-n_d}}{c_0 + c_1 q^{-1} + \dots + c_{n_c} q^{-n_c}}, \quad (31.2)$$

$$S(q) = \frac{B(q)}{A(q)} = \frac{b_0 + b_1 q^{-1} + \dots + b_{n_b} q^{-n_b}}{a_0 + a_1 q^{-1} + \dots + a_{n_a} q^{-n_a}}, \quad (31.3)$$

$$f(r(t)) = \sum_{j=0}^{n_w} w_j f_j(r(t)), \quad (31.4)$$

where  $q^{-1}$  denotes the backwards shift operator, and  $f_j$  forms a set of nonlinear basis functions. Without limitation, it is assumed in the remainder of the paper that these nonlinear basis functions are given by  $f_j(r) = r^j$ . Note that the method itself is not limited to a nonlinearity that is given by a nonlinear basis function expansion. Other nonlinearity representations, such as neural networks, could also be used in combination with the SADE optimisation based approach.

The Wiener-Hammerstein structure is subject to some identifiability issues since only the signals  $u(t)$  and  $y(t)$  are known. A gain exchange is possible in between the two LTI and the static nonlinear blocks. Also a delay exchange is possible in between the two LTI blocks. To obtain a unique model representation, the first nonzero coefficients (the nonzero coefficients belonging to the lowest power of  $q^{-1}$ ) of  $D(q)$ ,  $C(q)$ ,  $B(q)$ ,  $A(q)$  are set to zero, and all the full-sample delays are allocated to the LTI subsystem  $H(q)$ .

As a result, assuming that  $a_0$ ,  $b_0$ ,  $c_0$  and  $d_0$  are equal to units, the unknown parameter vector to estimate is given by:

$$\theta = [a_1 \dots a_{n_a} \ b_1 \dots b_{n_b} \ c_1 \dots c_{n_c} \ d_1 \dots d_{n_d} \ w_0 \dots w_{n_w}] \quad (31.5)$$

The model orders  $n_a$ ,  $n_b$ ,  $n_c$ ,  $n_d$  and  $n_w$  are set here by the user, although they could be determined by cross-validation in a free machine learning approach.

### 31.3 Evolutionary Algorithm-Based Identification

#### 31.3.1 SADE Algorithm

As it is often natural to frame system identification problems directly in terms of optimisation, it is thus natural to take advantage of the state of the art in optimisation. For some time now, evolutionary algorithms (EAs) have provided a powerful and versatile approach to optimisation and have therefore proved useful for SI. EAs began with the basic Genetic Algorithm (GA) and even the simplest form of that algorithm proved useful for SI; an early example of using a GA for the identification of Bouc-Wen hysteretic systems can be found in [21]. However, once real-parameter evolutionary schemes like Differential Evolution (DE) emerged [23], it quickly became clear that they offered major advantages for SI. The first application of DE for the Bouc-Wen model appeared in [24]. As in all evolutionary optimisation procedures, a population of possible solutions (here, the vector of parameter estimates), is iterated in such a way that succeeding generations of the population contain better solutions to the problem in accordance with the Darwinian principle of ‘survival of the fittest’. The problem is framed here as a minimisation problem with a least squares cost function defined as:

$$V(\theta) = \frac{1}{N} \sum_{t=1}^N (y(t) - \hat{y}(t, \theta))^2, \quad (31.6)$$

where  $N$  is the total number of samples in the estimation record, and  $y(t, \theta)$  is the modeled output given by Eq. (31.1) using the parameter set  $\theta$ .

The standard DE algorithm of reference [23] attempts to transform a randomly generated initial population of parameter vectors into an optimal solution through repeated cycles of evolutionary operations, in this case: *mutation*, *crossover* and *selection*. In order to assess the suitability of a certain solution, a cost or fitness function is needed; the cost function in Eq. (31.6) is the one used here. Figure 31.2 shows a schematic for the DE procedure for evolving between populations. The following process is repeated with each vector within the current population being taken as a *target vector*; each of these vectors has an associated cost taken from Eq. (31.6). Each target vector is pitted against a *trial vector* in a competition which results in the vector with lowest cost advancing to the next generation.

The mutation procedure used in basic DE proceeds as follows. Two vectors  $A$  and  $B$  are randomly chosen from the current population to form a vector differential  $A - B$ . A *mutated* vector is then obtained by adding this differential, multiplied by a scaling factor  $F$ , to a further randomly chosen vector  $C$  to give the overall expression for the mutated vector:  $C + F(A - B)$ . The scaling factor,  $F$ , is often found have an optimal value between 0.4 and 1.0.

The *trial vector* is the child of two vectors: the target vector and the mutated vector, and is obtained via a crossover process; in this work uniform crossover is used. Uniform crossover decides which of the two parent vectors contributes to each chromosome of the trial vector by a series of  $D - 1$  binomial experiments. Each experiment is mediated by a crossover parameter  $C_r$  (where  $0 \leq C_r \leq 1$ ). If a random number generated from the uniform distribution on  $[0,1]$  is greater than  $C_r$ , the trial vector takes its parameter from the target vector, otherwise the parameter comes from the mutated vector.

This process of evolving through the generations is repeated until the population becomes dominated by only a few low cost solutions, any of which would be suitable. Like the vast majority of optimisation algorithms, convergence to the global minimum is not guaranteed; however, one of the benefits of the evolutionary approach is that it more resistant to finding a local minimum.

A potential weakness of the standard implementation of the DE algorithm as described above is that it requires the prior specification of a number of *hyperparameters* (parameters which need to be specified before the algorithm can run). Apart from the population size, maximum number of iterations etc., the algorithm needs a priori specification of the scaling factor  $F$  and crossover probability  $C_r$ . The values used in [24] were chosen on the basis of trial and error; however, they are not guaranteed to work as well in all situations and an algorithm which establishes ‘optimum’ values for these parameters during the course of the evolution is clearly desirable. Such an algorithm is available in the form of the Self-Adaptive Differential Evolution (SADE) algorithm [19, 20]; the description and implementation of the algorithm here largely follows [20].

The development of the SADE algorithm begins with the observation that Storn and Price, the originators of DE, arrived at five possible strategies for the mutation operation [25]:

1. *rand1*:  $M = A + F(B - C)$
2. *best1*:  $M = X^* + F(B - C)$
3. *current-to-best*:  $M = T + F(X^* - T) + F(B - C)$
4. *best2*:  $M = X^* + F(A - B) + F(C - D)$
5. *rand2*:  $M = A + F(B - C) + F(D - E)$

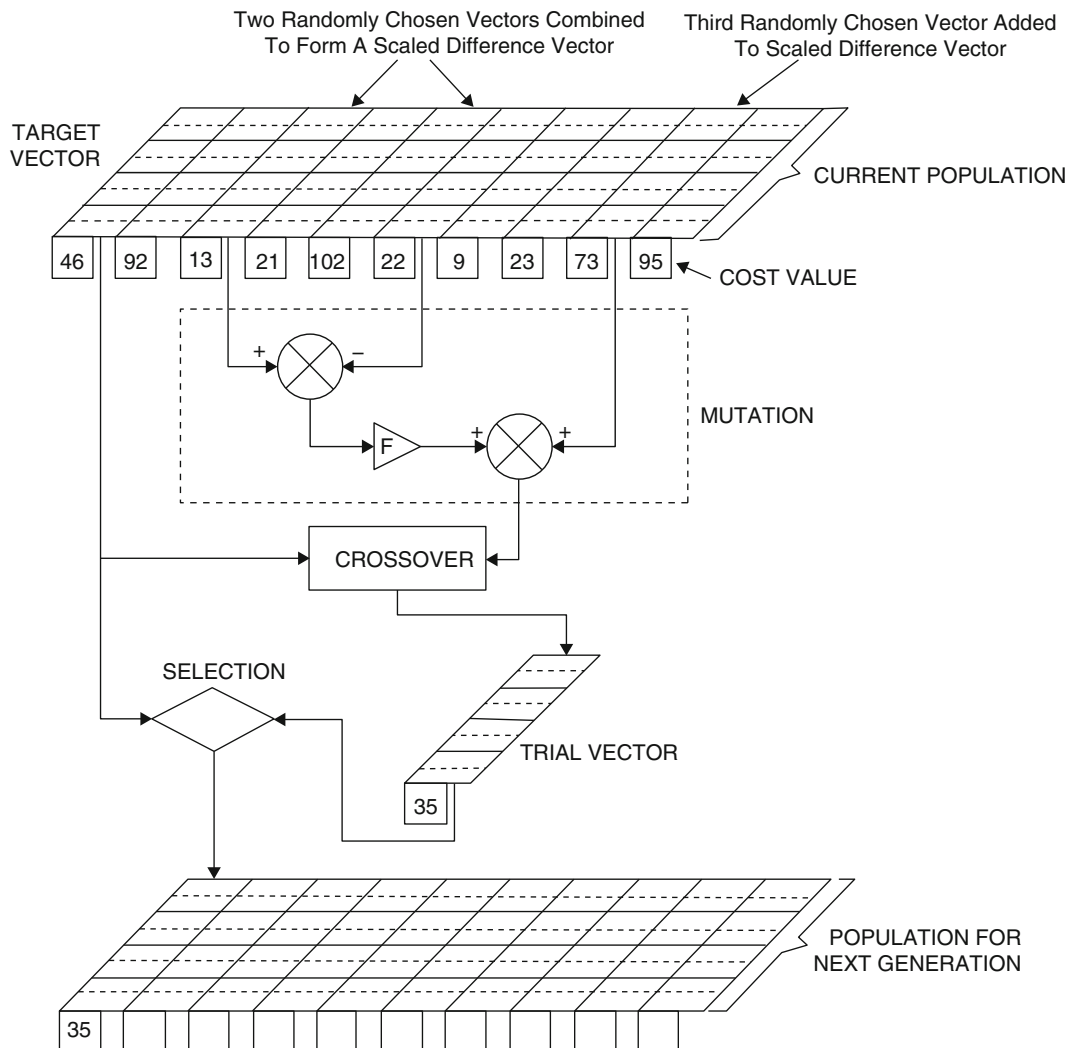


Fig. 31.2 Schematic for the standard DE algorithm

where  $T$  is the current trial vector,  $X^*$  is the vector with (currently) best cost and  $(A, B, C, D, E)$  are randomly-chosen vectors in the population distinct from  $T$ .  $F$  is a standard (positive) scaling factor. The SADE algorithm also uses multiple variants of the mutation algorithm as above; however these are restricted to the following four:

1. *rand1*
2. *current-to-best2*:  $M = T + F(X^* - T) + F(A - B) + F(C - D)$
3. *rand2*
4. *current-to-rand*:  $M = T + K(A - T) + F(B - C)$

In the strategy *current-to-rand*,  $K$  is defined as a coefficient of combination and would generally be assumed in the range  $[-0.5, 1.5]$ ; however, in the implementation of [20] and the one used here, the prescription  $K = F$  is used to essentially restrict the number of tunable parameters. The SADE algorithm uses the standard crossover approach, except that at least one crossover is forced in each operation on the vectors. If mutation moves a parameter outside its allowed (predefined) bounds, it is pinned to the boundary. Selection is performed exactly as in DE; if the trial vector has smaller (or equal) cost to the target, it replaces the target in the next generation.

The adaption strategy must now be defined. First, a set of probabilities are defined:  $\{p_1, p_2, p_3, p_4\}$ , which are the probabilities that a given mutation strategy will be used in forming a trial vector. These probabilities are initialised to be all equal to 0.25. When a trial vector is formed during SADE, a roulette wheel selection is used to choose the mutation strategy on the basis of the probabilities (initially, all equal). At the end of a given generation, the numbers of trial vectors successfully surviving to the next generation from each strategy are recorded as:  $\{s_1, s_2, s_3, s_4\}$ ; the numbers of trial vectors

from each strategy which are discarded are recorded as:  $\{d_1, d_2, d_3, d_4\}$ . At the beginning of a SADE run, the survival and discard numbers are established over the first generations, this interval is called the *learning period* (and is another example of a hyperparameter). At the end of the learning period, the strategy probabilities are updated by,

$$p_i = \frac{s_i}{s_i + d_i} \quad (31.7)$$

After the learning period, the probabilities are updated every generation but using survival and discard numbers established over a moving window of the last  $N_L$  generations. The algorithm thus adapts the preferred mutation strategies. SADE also incorporates adaption or variation on the hyperparameters  $F$  and  $C_r$ . The scaling factor  $F$  mediates the convergence speed of the algorithm, with large values being appropriate to global search early in a run and small values being consistent with local search later in the run. The implementation of SADE used here largely follows [19] and differs only in one major aspect, concerning the adaption of  $F$ . Adaption of the parameter  $C_r$  is based on accumulated experience of the successful values for the parameter over the run. It is assumed that the crossover probability for a trial is normally distributed about a mean  $\bar{C}_r$  with standard deviation 0.1. At initiation, the parameter  $C_r$  is set to 0.5 to give equal likelihood of each parent contributing a chromosome. The crossover probabilities are then held fixed for each population index for a certain number of generations and then resampled. In a rather similar manner to the adaption of the strategy probabilities, the  $C_r$  values for trial vectors successfully passing to the next generation are recorded over a certain greater number of generations and their mean value is adopted as the next  $\bar{C}_r$ . The record of successful trials is cleared at this point in order to avoid long-term memory effects. The version of the algorithm here adapts  $F$  in essentially the same manner as  $C_r$  but uses the Gaussian  $N(0.5, 0.3)$  for the initial distribution. At this point, the reader might legitimately argue that SADE has simply replaced one set of hyperparameters ( $F, C_r$ ) with another (duration of the learning period etc.). In fact, because DE and SADE are heuristic algorithms, there is no analytical counter to this argument. However, the transition to SADE is justified by the fact that the algorithm appears to be very robust with respect to the new hyperparameters.

From an SI point of view there are a number of advantages to the evolutionary approaches. First of all, in general, EAs are quite resistant to stalling in local minima because they use a (potentially large) population of possible solutions. Specific to SI problems, EAs offer the advantages that they work just as well for problems which are nonlinear in the parameters or have hidden or latent variables; one only needs measurements of any states which appear in the cost function. Of course, there are disadvantages too; the algorithms can be slow, depending on the computational cost of the objective function and, because the algorithms are fundamentally heuristic, there is no recourse to mathematics in order to prove theorems on parameter bias etc.

### 31.3.2 Initialisation

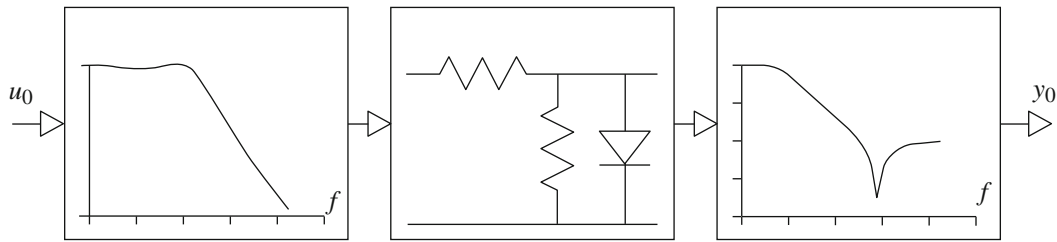
At the start of the SADE algorithm a random initial population is generated. This population is generated here such that the LTI subsystems  $H(q)$  and  $S(q)$  are stable, and such that the parameters  $\theta$  are limited within a given parameter range. The algorithm is implemented such that the parameters remain within that range during the optimisation.

## 31.4 Wiener-Hammerstein Benchmark Results

### 31.4.1 Benchmark Setup

A detailed description of the benchmark is given in [17]. The benchmark data are generated from a Wiener-Hammerstein nonlinear electronic circuit, as shown in Fig. 31.3. The first LTI block is a third order Chebyshev low-pass filter with 0.5 dB ripple and a cut-off frequency at 4.4 kHz. The second LTI block is a third order inverse Chebyshev low-pass filter with a  $-40$  dB stop band starting at 5 kHz. The static nonlinearity is a one-sided saturation implemented as a resistor-diode network. The system is excited by low-pass filtered Gaussian noise, with cut-off frequency set at 10 kHz. The input and output signals are measured with a sampling frequency equal to 51.2 kHz.

The benchmark setup is chosen as an illustration since a good comparison with other identification methods is possible using this system. A wide range of results on the 2009 Wiener-Hammerstein benchmark are reported in [26].



**Fig. 31.3** Wiener-Hammerstein benchmark system

**Table 31.1** Settings of the SADE optimisation algorithm

Population size	1600
Generation number	5000
Parameter range	[−55]
Update generation number	100
Mean update generation number	100
Strategy learning period	100

**Table 31.2** Results on the 2009 Wiener-Hammerstein benchmark

Run number	1	2	3	4	5	6	7	8	9	10
Estimation error rms (mV)	8.7	8.7	25.6	8.7	8.8	8.7	8.7	8.7	8.7	8.7
Estimation error mean (mV)	1.40	1.40	−1.81	1.40	1.43	1.40	1.40	1.40	1.40	1.40
Estimation error std (mV)	8.6	8.6	25.6	8.6	8.7	8.6	8.6	8.6	8.6	8.6
Validation error rms (mV)	9.3	9.3	32.0	9.3	9.5	9.3	9.3	9.3	9.3	9.3
Validation error mean (mV)	−0.33	−0.33	−0.50	−0.33	−0.36	−0.33	−0.33	−0.33	−0.33	−0.33
Validation error std (mV)	9.3	9.3	32.0	9.3	9.4	9.3	9.3	9.3	9.3	9.3

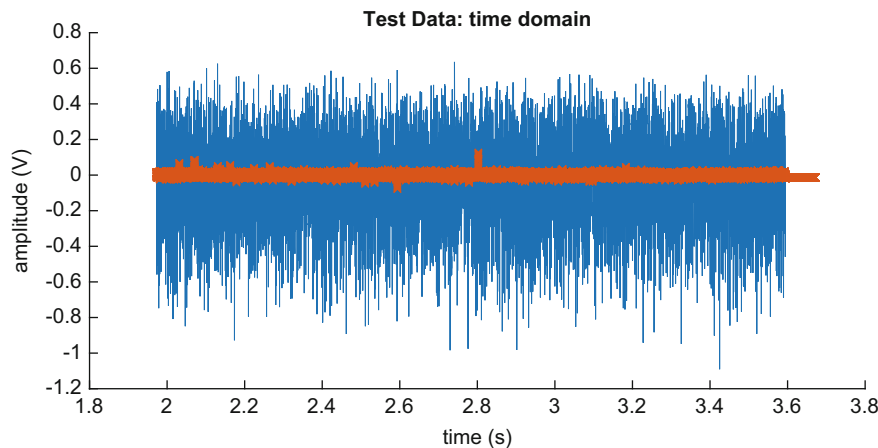
### 31.4.2 Model Estimation

The estimation of the model is performed on a small set of the available estimation data, only samples 4901–6000 are considered. The model orders  $n_a$ ,  $n_b$ ,  $n_c$ ,  $n_d$  are set equal to 3, the static nonlinearity is estimated as a third degree polynomial. This results in a total of 16 free parameters to estimate.

The SADE algorithm is set to use a population size of 1600 and runs for 5000 iterations. The parameter values are limited to the range [−5, 5], the exact settings of the SADE algorithm can be found in Table 31.1. To test the robustness of the proposed algorithm, ten different runs of the algorithm are made, each starting from a new, randomly generated, initial population.

### 31.4.3 Validation Results

The validation estimation results are reported in Table 31.2, as requested in [17] the root mean square error (RMSE) is used as the error criterion. The obtained RMSE is comparable with the one obtained by other Wiener-Hammerstein identification methods using a third-order polynomial static nonlinearity [26]. The time-domain validation output and the obtained model error are shown in Fig. 31.4. Note that the peaks in the error suggests that a further improvement of the RMSE can be expected if a higher order polynomial nonlinearity, or another nonlinearity structure is considered. Note that the proposed approach converges in 8 out of the 10 cases to the same minimum. Run number 5 is very near to this minimum, and after a closer inspection it turned out that this particular run was not yet fully converged to its minimum. Run number 3 got stuck in a local minimum.



**Fig. 31.4** Time-domain representation of the measured validation output (*blue*) and the model error (*red*)

## 31.5 Conclusions

This paper illustrates how evolutionary algorithms in general, and the SADE algorithm in particular, offer a robust and user-friendly optimisation tool for the identification of block-oriented structures. The Wiener-Hammerstein structure is studied in detail in this paper, however, one can expect to obtain similar results on other model structures.

The SADE evolutionary algorithm used in this paper requires almost no user interaction. It is known to be quite robust with respect to the setting of the hyperparameters. The main downside of the evolutionary algorithms is the rather heavy computational load.

**Acknowledgements** Maarten Schoukens would like to gratefully acknowledge the support of the Fund for Scientific Research (FWO), the Methusalem grant of the Flemish Government (METH-1), the IAP VII/19 DYSCO program, and the ERC advanced grant SNLSID under contract 320378.

Keith Worden would like to gratefully acknowledge the support of the UK Engineering and Physical Sciences Research Council (EPSRC) through grant reference numbers EP/J016942/1 and EP/K003836/2.

## References

- Giri, F., Bai, E.W. (eds.): Block-Oriented Nonlinear System Identification. Volume 404 of Lecture Notes in Control and Information Sciences. Springer, Berlin/Heidelberg (2010)
- Vandersteen, G., Rolain, Y., Schoukens, J.: Non-parametric estimation of the frequency-response functions of the linear blocks of a Wiener-Hammerstein model. *Automatica* **33**(7), 1351–1355 (1997)
- Tiels, K., Schoukens, M., Schoukens, J.: Initial estimates for Wiener-Hammerstein models using phase-coupled multisines. *Automatica* **60**, 201–209 (2015)
- Weiss, M., Evans, C., Rees, D.: Identification of nonlinear cascade systems using paired multisine signals. *IEEE Trans. Instrum. Meas.* **47**(1), 332–336 (1998)
- Tan, A.H., Godfrey, K.: Identification of Wiener-Hammerstein models using linear interpolation in the frequency domain (LIFRED). *IEEE Trans. Instrum. Meas.* **51**(3), 509–521 (2002)
- Ben Ahmed, Z., Favier, G., Derbel, N.: Tensor-based methods for Wiener-Hammerstein system identification. In: 10th International Multi-conference on Systems, Signals Devices (SSD), 2013, Niagara Falls, pp. 1–6, Mar 2013
- Dewhurst, O.P., Simpson, D.M., Angarita, N., Allen, R., Newland, P.L.: Wiener-Hammerstein parameter estimation using differential evolution: application to limb reflex dynamics. In: International Conference on Bio-inspired Systems and Signal Processing, Valencia, pp. 271–276 (2010)
- Schoukens, M., Vandersteen, G., Rolain, Y., Ferranti, F.: Fast identification of Wiener-Hammerstein systems using discrete optimization. *IET Electron. Lett.* **50**(25), 1942–1944 (2014)
- Naitali, A., Giri, F.: Wiener-Hammerstein system identification – an evolutionary approach. *Int. J. Syst. Sci.* **47**(1), 45–61 (2016)
- Pintelon, R., Schoukens, J.: System Identification: A Frequency Domain Approach, 2nd edn. Wiley/IEEE Press, Hoboken (2012)
- Schoukens, J., Vaes, M., Pintelon, R.: Linear system identification in a nonlinear setting: nonparametric analysis of the nonlinear distortions and their impact on the best linear approximation. *IEEE Control Syst.* **36**(3), 38–69 (2016)
- Billings, S.A., Fakhouri, S.Y.: Identification of a class of nonlinear systems using correlation analysis. *Proc. Inst. Electr. Eng.* **125**(7), 691–697 (1978)



13. Sjöberg, J., Lauwers, L., Schoukens, J.: Identification of Wiener-Hammerstein models: two algorithms based on the best split of a linear model applied to the SYSID'09 benchmark problem. *Control Eng. Pract.* **20**(11), 1119–1125 (2012)
14. Schoukens, M., Pintelon, R., Rolain, Y.: Identification of Wiener-Hammerstein systems by a nonparametric separation of the best linear approximation. *Automatica* **50**(2), 628–634 (2014)
15. Vanbeylen, L.: A fractional approach to identify Wiener-Hammerstein systems. *Automatica* **50**(3), 903–909 (2014)
16. Giordano, G., Sjöberg, J.: A time-domain fractional approach for Wiener-Hammerstein systems identification. In: 17th IFAC Symposium on System Identification SYSID 2015, Beijing, vol. 48, no. 28, pp. 1232–1237 (2015)
17. Schoukens, J., Suykens, J.A.K., Ljung, L.: Wiener-Hammerstein benchmark. In: 15th IFAC Symposium on System Identification (SYSID), Saint-Malo, July 2009
18. Schoukens, M., Noël, J.P.: Wiener-Hammerstein benchmark with process noise. In: Workshop on Nonlinear System Identification Benchmarks, Brussels, pp. 15–19, Apr 2016
19. Qin, A.K., Suganthan, P.N.: Self-adaptive differential evolution algorithm for numerical optimization. In: 2005 IEEE Congress on Evolutionary Computation, Edinburgh, pp. 1785–1791 (2005)
20. Huang, V.L., Qin A.K., Suganthan, P.N.: Self-adaptive differential evolution algorithm for constrained real-parameter optimization. In: IEEE Congress on Evolutionary Computation, Vancouver, pp. 17–24 (2006)
21. Deacon, B.P., Worden, K.: Identification of hysteretic systems using genetic algorithms. In: Proceedings of EUROMECH – 2nd European Nonlinear Oscillations Conference, Prague, pp. 55–58 (1996)
22. Worden, K., Manson, G.: On the identification of hysteretic systems. Part I: fitness landscapes and evolutionary identification. *Mech. Syst. Signal Process.* **29**(0), 201–212 (2012)
23. Price, K., Storn, R.: Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. *J. Glob. Optim.* **11**(4), 341–359 (1997)
24. Kyprianou, A., Worden, K., Panet, M.: Identification of hysteretic systems using the differential evolution algorithm. *J. Sound Vib.* **248**(2), 289–314 (2001)
25. Storn, R., Price, K.: Technical report. <http://www.icsi.berkeley.edu/~storn/code.html>. Accessed 27 oct 2009
26. Marconato, A., Schoukens, M., Rolain, Y., Schoukens, J.: Study of the effective number of parameters in nonlinear identification benchmarks. In: 52nd IEEE Conference on Decision and Control, Florence, pp. 4308–4313, Dec 2013,