MODELING AND SIMULATION OF WIND ENERGY PRODUCTION IN THE SMART-GRID SCENARIO

Laura Pérez-Vilarelle

Dept. of Computer Architecture and Automation
Complutense University of Madrid
Madrid, Spain
laurap12@ucm.es

José L. Risco-Martín

Dept. of Computer Architecture and Automation
Complutense University of Madrid
CCS-Center for Computational Simulation
Madrid, Spain
jlrisco@ucm.es

José L. Ayala

Dept. of Computer Architecture and Automation
Complutense University of Madrid
CCS-Center for Computational Simulation
Madrid, Spain
jayala@ucm.es

ABSTRACT

Renewable energies, in particular wind energy, are characterized as highly variable and unpredictable in terms of production, and they are increasingly more important in the context of the smart grid energy production. In this scenario, accurate prediction models and techniques are desirable to optimize the renewable energy production and reduce the environmental impact. In this article, we propose the development of predictive techniques based on mathematical models, and the integration in a simulation framework that enables the simulation of variable conditions in wind energy production. The system also offers the possibility to automatically select the most reliable model for the current conditions. Our results show an accuracy of prediction (model fit) of up to 84%. The proposed simulation framework has been stressed with real data acquired from wind turbines in the area of Spain, providing efficient model selection and tuning of optimization parameters.

Keywords: ARIMA, N4SID, DEVS, renewable energy, smart-grid.

1 INTRODUCTION

Fossil fuels, which include coal, natural gas, petroleum, shale oil, and bitumen, are the main sources of heat and electrical energy. All these fuels contain – besides the major constituents (carbon, hydrogen, oxygen) – other materials including metal, sulfur and nitrogen compounds. Gross emission of pollutants is tremendous all over the world. These pollutants are present in the atmosphere in such conditions that they can affect man and his environment.

SpringSim-MSCIAAS, 2018 April 15-18, Baltimore, MD, USA;

Those higher consumers of electrical energy are also those with a the higher environmental impact. For example, computing data centers consume about 2% of the worldwide energy production and pollute with more than 43 million tons of CO_2 generated per year (Arroba et al. 2015).

In this scenario, the continuous growth of renewable energies, from 8.5% in 2004 up to 16.7% in 2015 in EU, (Eurostat 2017), is a trend that helps in the system balance but requires to make full use of a smart electrical grid. The European Technology Platform Smart Grid, (ETPSG), defines the smart grid as an electricity network that can intelligently integrate the actions of all users connected to it: generators, consumers and those that do both in order to efficiently deliver sustainable, economic and secure electricity supplies.

Optimizing the smart grids constitutes a great challenge. Two main problems can be identified : i) managing variability during the continuous balancing of the system; and ii) balancing energy supply and demand under scarcity and surplus situations derived from the demand of huge consumers.

The generators of renewable energy suffer from variance on atmospheric conditions along the time, causing the discontinuity and variability of the production of renewable energy, and injecting the system with a discontinuous energy flow. Thus, a good prediction of energy generation can optimize its use in the context of smart grid.

On the other hand, the consumers, in particular high electrical consumers like data centers or smart centers, play a key role in the need for optimization of the smart grid. Only in the US, data centers consumed around $20 billion (200 TWh) yearly of electricity in 2016, and this amount doubles every five years (Tomes and Altiparmak 2017). Therefore, the goal of the research presented in this paper is the development of prediction models and techniques for renewable energies in a smart grid environment.

To achieve the aforementioned objective, we will develop a simulation framework that will allow:

- To analyze and visualize the generation vs. consumption status at every instant.
- To provide the scalability needs in the grid system and evaluate their impact on performance.
- To design the smart grid in base to the optimum.

The described system will integrate the models derived for prediction of renewable energy production. The simulation system will be designed in such a way that it will be easily extended with the models for the remaining agents in the smart grid scenario: large energy consumers, traditional energy generation and energy storage units.

This paper is structured as follows. Section 2 describes two of the predictive models used in the wind energy simulation system (N4SID and ARIMA). It also presents the Discrete Event System Specification, with which the energy prediction system will be simulated. Section 3 is structured in two parts, first part describes the modeling of energy prediction whereas second part explains the simulation system. Then, section 4 presents the experimental results obtained and finally in section 5 the conclusions obtained are drawn as well as some future work proposals.

2 BACKGROUND

In the previous section, the importance of modeling, simulation and forecasting of renewable energies in a smart grid environment has been motivated; however, this is a complex problem that presents operational difficulties due to the intermittent and highly variable nature of the energy sources. There are many studies (Mao and Shaoshuai 2016), (Ambach and Vetter 2016), (Tan et al. 2016) on wind energy prediction that support the possibility of modeling it with different methods such as regression analysis, Kalman filtering, Autoregressive Integrated Moving Average (ARIMA), state space models, artificial intelligence, neural networks, fuzzy inference methods, genetic programing or support vector regression among others.

In the work presented here, we have selected two classical algorithms that offer the required reliability in the prediction, and the sufficient prediction horizon. These algorithms implement different modeling approaches (state space and time series) to give generality to the predictor system.

On the other hand, the wind energy generation subsystem, in which predictive models are included, is implemented in a simulation environment that will allow us to achieve the mentioned objectives and to tackle the global system and future research lines with guarantees. In other words, we design a simulation environment that allows us to incorporate models for energy production, energy demand, and energy storage in a flexible and scalable way. Moreover, the simulator provides the required control capability to select the best model in real time.

2.1 N4SID: Numerical algorithms for subspace state space system identification

N4SID is the acronym of Numerical algorithms for subspace state space system identification (Overschee and de Moor 1993). This algorithm is a type of subspace identification method and estimates a linear time invariant state space model from input-output measurements.

Consider the general linear discrete time invariant state space model:

$$\begin{aligned} x_{k+1} = &\quad A_k x_k + B_k u_k + K_k e_k. \\ y_k = &\quad C_k x_k + D_k u_k + e_k. \end{aligned} \tag{1}$$

Where $x_k \in \mathbb{R}^n$ is the system state, $y_k \in \mathbb{R}^l$ represents the outputs and $u_k \in \mathbb{R}^m$ the inputs. While $K_k e_k \in \mathbb{R}^n$ and $e_k \in \mathbb{R}^l$ are additional unknown noise sequences.

The goal is to estimate the order n of the unknown system and the system matrices $A \in \mathbb{R}^{nxn}$, $B \in \mathbb{R}^{nxm}$, $C \in \mathbb{R}^{lxn}$ and $D \in \mathbb{R}^{lxm}$. It first approximates the state sequence of the system and then uses the approximate state in a second step to estimate the system matrices.

The mathematic identification system, N4SID, uses geometric, matrix and algebraic concepts that eliminate optimal problems in local and iterative processes, reducing the computational cost. The key of this method is the oblique projection of subspaces generated by the block Hankel matrices formed by input and output data of systems.

2.2 ARIMA: Autoregressive integrated moving average

AutoRegressive Integrated Moving Average, (ARIMA), is a dynamic model of time series which combines integrated with autoregressive and a moving average model (Box et al. 2008). It was developed in the late sixties of the twentieth century. Box and Jenkins (1976) systematized it.

Autoregressive model, $AR(p)$, means that we forecast the variable of interest using a linear combination of past values of the variable, it is a regression of the variable against itself. This is like a multiple regression but with lagged values of y_t as predictors. A moving average model, $MA(q)$, uses past forecast errors in a regression like model. In this case each value of y_t can be thought of as a weighted moving average of the past few forecast errors. If we combine differencing with all previous, an ARIMA model is obtained, (integration in this context is the reverse of differencing).

The ARIMA model can be written as:

$$y'(t) = c + \phi_1 y'_{t-1} + \cdots + \phi_p y'_{t-p} + \theta_1 e_{t-1} + \cdots + \theta_q e_{t-q} + e_t. \tag{2}$$

We call this an $ARIMA(p,d,q)$ model, where $y'(t)$ is the differenced series, ϕ and θ are the autoregressive and moving average polynomials respectively, p and q are the order of autoregressive and moving average parts and d degree of first differencing involved. The number of past observations that y_t depends on, p, is the *AR* degree and the number of past innovations that y_t depends on, q, is the *MA* degree.

In the Box Jenkins methodology, a differencing approach is used to stable the original data and it uses autocorrelation and partial autocorrelation to decide which component should be included in the ARIMA model. Therefore, implementing an ARIMA model is analogous to differentiate the data series until it becomes stationary and, then, model the data with an ARMA model.

The general expression of ARMA is given by

$$y(t) = c + \phi_1 y_{t-1} + \cdots + \phi_p y_{t-p} + \theta_1 e_{t-1} + \cdots + \theta_q e_{t-q} + e_t,$$

where ϕ and θ are the autoregressive and moving average polynomials respectively.

In the case that there were more than one exogenous variable, the previously described models are generalized and receive the name of ARIMAX and ARMAX respectively. The expression for this ARMAX model is as follows:

$$\begin{aligned} y(t) + a_1 y(t-1) + \cdots + a_{n_a} y(t-n_a) = \quad & b_1 u(t-n_k) + b_{n_b} u(t-n_k-n_b+1) + \\ & c_1 e(t-1) + \cdots + c_{n_c} e(t-n_c) + e(t). \end{aligned} \tag{3}$$

And a more compact way to write this equation is:

$$A(q)y(t) = B(q)u(t-n_k) + C(q)e(t). \tag{4}$$

where $y(t)$ is the output at time t, n_a, n_b and n_c are number of poles, zeroes plus 1, and C coefficients respectively. n_k is the number of input samples that occur before the input affects the output. Finally $y(t-1) + \cdots + y(t-n_a)$ are the previous outputs on which the current output depends, $u(t-n_k) + u(t-n_k-n_b+1)$ are the previous and delayed inputs on which the current output depends, and $e(t-1) + \cdots + e(t-n_c)$ is the white-noise disturbance value. The parameters n_a, n_b, and n_c are the orders of the ARMAX model, and n_k is the delay. q is the delay operator, while A, B and C parameters can be obtained from:

$$\begin{aligned} A(q) &= 1 + a_1 q^{-1} + \cdots + a_{n_a} q^{-n_a} \\ B(q) &= b_1 + b_2 q^{-1} + \cdots + b_{n_b} q^{-n_b+1} \\ C(q) &= 1 + c_1 q^{-1} + \cdots + c_{n_c} q^{-n_c}. \end{aligned}$$

2.3 The Discrete Event System Specification

The Discrete Event System Specification (DEVS) is a general formalism for discrete event system modeling based on set theory (Zeigler, Praehofer, and Kim 2000). The DEVS formalism provides the framework for information modeling which gives several advantages to analyze and design complex systems: completeness, verifiability, extensibility, and maintainability. Once a system is described in terms of the DEVS theory, it can be easily implemented using an existing computational library. The parallel DEVS (PDEVS) approach was introduced, after 15 years, as a revision of Classic DEVS. Currently, PDEVS is the prevalent

DEVS, implemented in many libraries. In the following, unless it is explicitly noted, the use of DEVS implies PDEVS.

There exist two DEVS models types: atomic and coupled. Atomic models are directly expressed using three sets and five functions. The formal definition of a atomic model is described as:

$$A = \langle X, Y, S, \delta_{\text{ext}}, \delta_{\text{int}}, \delta_{\text{con}}, \lambda, ta \rangle \tag{5}$$

where (X), (Y), (S) are the input, output and state sets respectively and (δ_{ext}), (δ_{int}), (δ_{con}), (λ) and (ta) are external transition, internal transition, confluent, output and time advance functions respectively.

Atomic DEVS processes input events based on their model's current state and condition, generates output events and transition to the next state.

The coupled model is the aggregation/composition of two or more atomic and coupled models connected by explicit couplings. They are expressed using three sets and three coupling relations, the formal definition of a coupled model is:

$$M = \langle X, Y, C, EIC, EOC, IC \rangle \tag{6}$$

where X and Y are the sets of inputs and outputs respectively, C is a set of DEVS component models (atomic or coupled) and EIC, EOC, IC are input, output and internal coupling relations.

Given the recursive definition of M, a coupled model can itself be a part of a component in a larger coupled model system giving rise to a hierarchical DEVS model construction.

3 FRAMEWORK

The proposed framework consists of two main parts based on the approaches presented in the previous section: (a) definition of the predictive models for energy generation, and (b) the discrete event-based framework for the integration and simulation of the whole environment.

3.1 Predictive models

The predictive models are computed to obtain a representative model of the energy generated by the renewable energy source, in order to predict wind generation within a time window based on meteorological monitoring data. The final framework contains a set of predictive models obtained using different techniques, selecting the best one in real time. In this work, we have computed two predictive models using N4SID and ARIMA.

To obtain each predictive model, we split the data used for the modeling into three sets: training, validation and test. The training stage trains these models (described by equation (1) and equation (2)) using identification methods, comparing the output of the model with the expected actual output. After that, during the validation stage we estimate how well the model has been trained. Finally, the test stage is the one used to provide an unbiased evaluation of the final model fit from the training dataset.

The fit used to evaluate both ARIMA and N4SID models is the Normalized Root Mean Squared Error (NRMSE), which measures how well the response of the model fits the estimation data. It is expressed as a percentage, and defined as:

$$Fit = 100 * (1 - \frac{||y - \hat{y}||}{||y - \bar{y}||}), \tag{7}$$

where y is the actual output data, $\bar{y}$ is the mean, $\hat{y}$ is the predicted response of the model and $||.||$ indicates the 2-norm of a vector.

In our case, the predicted wind speed v and wind direction d are the input data for our prediction model, and the energy produced is the output data obtained from the model.

In the case of N4SID, we have fixed a future horizon and we have trained the model with different past horizons. Then we have selected the model with the best adjustment in the validation stage, and thus, we have found the best model quality in the test stage.

In the case of ARIMA, we have selected a set of possible values for p, d and q, and iterated the *armax* function of Matlab (MATLAB 2015) against the training data set. We have compared the different obtained models for the training and validation sets, and selected the model with the best fit.

3.2 Discrete event simulation framework

As mentioned before, it is of great relevance to be able to include the generation of wind energy in a simulation environment. To target this objective, a simulation system was implemented in DEVS that includes the wind energy predictive models computed above.

The simulation process includes: (a) access to the required data, (b) predictive models that perform the calculation/prediction of the energy, (c) a decision manager, which based on the results obtained for the different models selects best one at each moment, and (d) modules to compute again the different predictive models in case they reach an error offset.

As a result, the simulation framework is able to emulate the final wind energy production model, where each model can be trained again if necessary.

In the following, we define the structure of our DEVS model. The root coupled model is composed of the following atomic models: Actual Energy, Predicted Input Data, Predictive Model 1, Predictive Model 2, ..., Predictive Model n, Decisor, Training and Validation Module 1, Training and Validation Module 2, ..., Training and Validation Module n, as can be seen in Figure 1.

The purpose of these atomic models is as follows:

- **Predicted input data:** It is an atomic model and its task is to read values of different predicted variables such as wind speed or wind direction at each time step. The output of this component will be used by different atomic models. These data typically come from a weather station.
- **Actual Energy:**This atomic model takes the actual energy provided by the wind turbine.
- **Model** 1**, Model** 2**,** ... **Model** n**:** These are different atomic models to obtain predicted values of energy as function of wind speed, wind direction, etc., and trained as explained in the previous Section, each one following a different technique.
- **Decisor:** This atomic model has been designed to decide the best model to use. To this end, it computes the fit of the last 24 hours for each model and uses the best one to select the current output. On the other hand, if the error of a Model i with respect to the actual energy is greater than an offset, then the `Decisor` activates the *Training and Validation i* module to train the model i again. The inputs to the `Decisor` atomic model are the different predictions provided by the models and the real value of the energy.
- **Training and Validation** 1**, Training and Validation** 2**, ..., Training and validation** n**:** These atomic models are used to train the different predictive models used in the system.

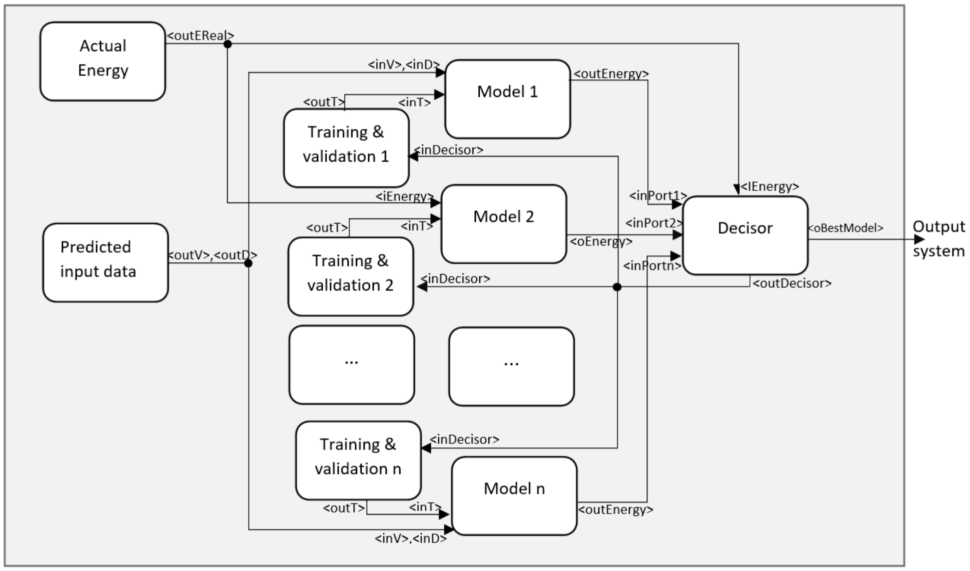


Figure 1: Coupled system for discrete event simulation, composed of atomic models.

In particular, we have incorporated three predictive models (Model 1, Model 2 and Model 3): ARIMA, N4SID and the Zero Order Hold (ZOH) model. ZOH has been included here as a simplistic model to evaluate the wellness and accuracy of the proposed N4SID and ARIMA techniques.

In the case of N4SID, the model uses as inputs the speed and direction of wind. It also receives all the matrices given by (1), as soon as the training and validation module finishes its training phase.

Regarding ARIMA, it needs as inputs the N past values of the predicted energy, wind speed and direction (being N determined by the order of the model, n_a, n_b and n_c). This model uses these values to solve the time series equation (3) and to provide the value of energy. As N4SID, this atomic model also can receive A, B and C vectors (defined by equation (4)) and the order of the model from the training and validation module.

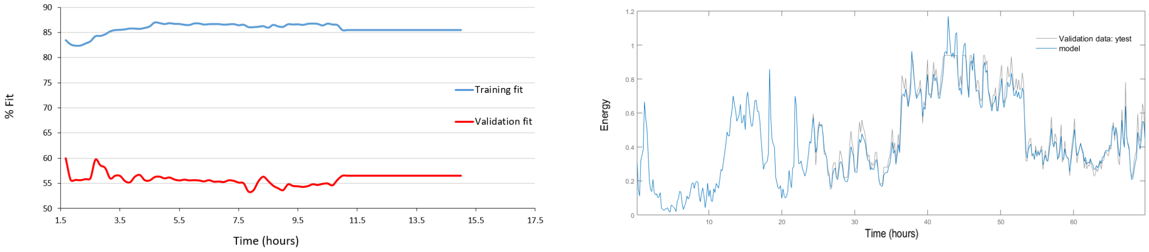
Finally, in the case of ZOH the input for obtaining the wind energy prediction is the previous value of the actual energy in a fixed horizon. The model simply retrieves the value collected in a past horizon and converts it to the current value.

4 EXPERIMENTAL RESULTS

In this section, we present the obtained experimental results. In Section 4.1, we describe the method followed to obtain our predictive models. Next, in Section 4.2, we show the results when those models are placed in a real context using the DEVS global model.

4.1 Predictive models

In order to properly train the models, we need: (a) wind speed and direction, and (b) the actual energy used in the identification model. On the one hand, wind speed and direction have been obtained from a wind farm located at Galicia, Spain, published by (Sotavento Galicia, S.A. 2017). This dataset is composed of 1677 samples, which correspond to the wind speed and direction measured every 10 minutes during 12 days. On the other hand, because of the impossibility to access to the real measures of a single wind turbine,



(a) (b)

Figure 2: (a) N4SID fit results in training and validation phases using different past horizons. (b) Best N4SID model response. Time response comparison between the selected model and actual energy data in the test phase.

the actual energy is computed using a three-phase asynchronous wind turbine generator available inside the Simscape Power Systems Matlab/Simulink toolbox (MATLAB 2015). This generator transforms wind speed and direction to turbine output power. When the wind speed is below the cut-in speed or above the cut-out speed, the machine generates zero real power. The machine always consumes reactive power. The local load consumes 75 KW. When the turbine generator produces more than 75 KW, the excess power is then exported to the grid. We use this generator as the golden standard because the database provided by Sotavento includes the total energy produced by the farm, which is not adequate for the purposes of this work.

As stated in previous Sections, we have trained two power models: N4SID and ARIMA. In both cases, training, validation and test sets represent a 50%, 25% and 25% of the total dataset, respectively.

4.1.1 N4SID model

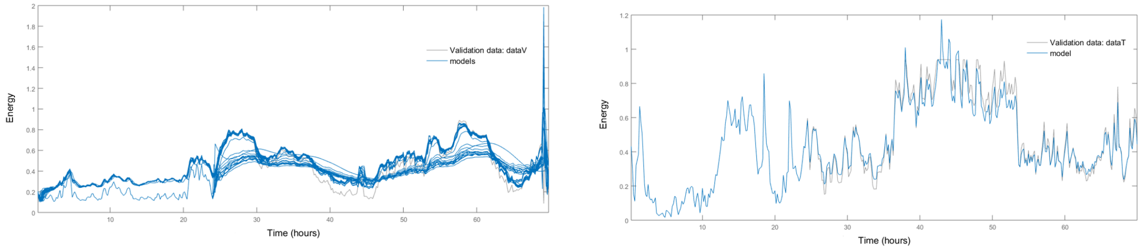
The MATLAB N4SID function estimates the best order state space model, and in particular all the A, B, C, and D matrices defined in Section 2, represented in equation (1).

Since we are predicting energy production for the next 24 hours, we have set the future horizon to this value. With this horizon fixed, we have generated models for different past horizons. The adjustment of the model to the actual data has been calculated using (7), both in the training and validation phases. Figure 2(a) shows the evolution of this process. In the training phase, a maximum value of 86.95% was reached. However, the final model is selected in the validation phase. In this case, the best fit was 60%, obtained for a past horizon of 1.5 hours or equivalently 10 previous past outputs. The fit is lower than training phase because the data set follow a different trend in this phase, but this fact is not relevant in the identification model.

Figure 2(b) depicts the time response of the previous N4SID selected model, using the test dataset. Obviously, as Figure 2(b) shows, the responses of the first 24 hours are identical, because these two signals are in the selected past horizon. Beyond that limit, we can appreciate how the model captures the trend quite well, with a fit value of 84.91%.

4.1.2 ARIMA model

Similarly to N4SID, the ARIMA MATLAB algorithm obtains the results of A, B and C vectors, all detailed in Section 2.2.



(a) (b)

Figure 3: (a): Time response comparison between candidate ARIMA models and actual energy produced in validation phase. (b) Best ARIMA model. Time response comparison between the selected model and actual energy produced in the test phase.

To find the best values for p, d and q, we have performed a stationary analysis of mean and variance over the dataset, with correlation simple and partial. In this case, we have obtained $2 \leq p \leq 5$, and $0 \leq q,d \leq 2$, or equivalently, $2 \leq n_a \leq 5, 1 \leq n_b \leq 2, 1 \leq n_c \leq 2, 0 \leq n_k \leq 2$. After that, we have iterated among these values using the ARMAX MATLAB function for the training dataset. Finally, we have compared these 48 different models in the training and validation set and as with the N4SID model we have selected the model with the best fit in the validation stage.

Figure 3(a) depicts the time responses (in validation) of all the models generated with the ARMAX function and using the previous set of (p,d,q) possible values. As Figure 3(a) shows, all the model capture the trend of the actual energy produced. These models show fit values in the range $[32.91\%, 52.78\%]$.

As with the N4SID model, Figure 3(b) shows the time response, in the test dataset, of the ARIMA model selected in the previous phase against the actual energy produced. As can be seen, the adjustment is perfectly admissible, reaching a fit of 84.23%. It proofs that the parameters selected are adequate, even when the validation dataset follows a different profile than the training dataset.

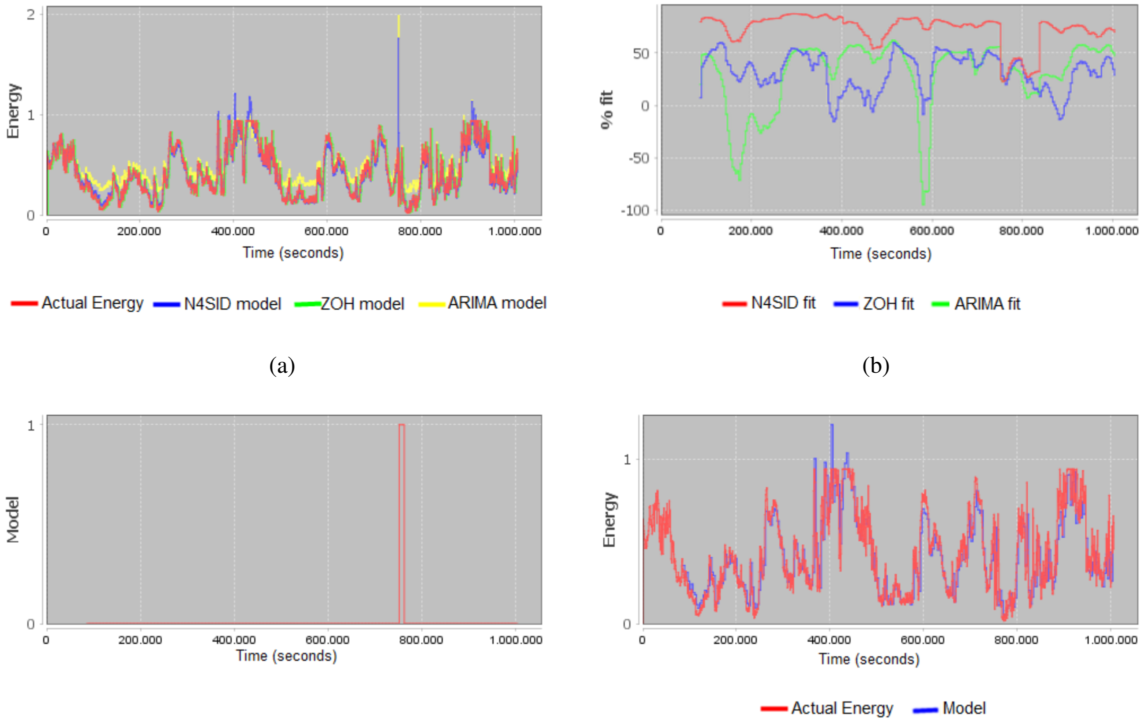
In the following Section we show the results of these two models when are used in a real context simulated through the framework presented in Figure 1. As expected both fits will be lower, since in the real-time simulation, we will not be able to compute the best initial state as MATLAB internally does when using the `compare` function.

4.2 Wind energy simulation

The simulation model proposed in Figure 1 has been implemented in DEVS using the xDEVS Java simulation toolkit (Risco-Martín et al. 2017).

To perform the simulation, we have used the total dataset described above. These data represent 1677 inputs over a total time of 12 days. To compare the performance of both N4SID and ARIMA models, we have included the *Zero Order Hold (ZOH)* model, which holds the current actual energy for a predefined time window. We have set this time window to 1 hours. As a consequence, the energy predicted by the ZOH model for the instant time t, $\hat{E}(t)$ is the actual energy produced 1 hours ago, $E(t-1)$. As future work, we are planning to include more predictive models, including supervised machine learning classification and regression techniques.

Each predictive model of the simulation framework depicted in Figure 1 receives as input, depending on each case, wind speed prediction, wind direction prediction, past energy values, etc. The actual energy



(a) (b)

(c) (d)

Figure 4: (a) N4SID, ARIMA and ZOH predictions against actual energy. (b) Fits obtained by the different models during the whole simulation. (c) Model selected at each time instant. 0 means that the N4SID model has been selected, 1 ZOH, and 2 ARIMA. (d) Simulation of joint model (N4SID+ZOH+ARIMA). Actual energy and values of predictive model selected for each time.

is used to compare with predictions. To prove the correctness of our proposed architecture, we are using those values (wind speed and direction as well as actual energy) already taken to train, validate and test our models. Then, each predictive model (N4SID, ARIMA and ZOH) is in charge of generating the respective estimated Energy $\hat{E}$ value. The decision system takes the best value between all the predictive models.

Figure 4(a) represents the output of each predictive model, i.e., N4SID, ARIMA and ZOH, as well as the actual energy collected. As can be seen, all the three models perform very well, as stated in the training phases described above. As Figure 4(a) shows, the ARIMA model is most of the time above the actual energy value, which will negatively impact on the final ARIMA fit value.

Figure 4(b) shows the evolution of the fit parameter obtained by the three predictive models in simulation runtime, using (7). This parameter cannot be computed until the second "virtual" day of the simulation, where both predicted and actual energy can be compared. The main difference with the fit in training, validation and test phase is that this framework is simulating a real case, where the best initial state is not computed at all. As a result, the fit offered by the predictive models is worse in the real context simulation, 72.37% for N4SID, 29.84% for ARIMA and 31.93% for ZOH. As can be seen N4SID is quite robust, independently of the initial conditions in the simulation.

Figure 4(c) shows the output of the `Decisor` atomic model. The behavior is like a multiplexer, where a model number is selected (0 for N4SID, 1 for ZOH and 2 for ARIMA) according the the best fit obtained in the last time window of 24 hours. As can be seen, N4SID is always dominant, with the exception of a little window at the end of the 8th day (between 8.7 and 8.8), where the model is changed to ZOH. Particularly, system includes certain conditions (for example, to set the number of instants necessary to recalculate the models adjustment) avoiding alternation of models during short periods of time.

Finally, Figure 4(d) shows the output of the joint model (N4SID+ZOH+ARIMA), which is the output of the `Decisor` atomic model. This value is compared with the actual energy. The fit of the joint model is computed as well, obtaining a value of 72.45%, which is obviously greater than all the N4SID, ZOH and ARIMA values, 72.37%, 31.93% and 29.84%, respectively.

5 CONCLUSIONS

The efficient optimization of renewable energies in the context of smart-grid production requires the development of accurate predictive techniques and models, and the provisioning of smart simulation frameworks that integrate such models and are able to automate decisions.

In this work, we have provided models for a realistic wind turbine in the region of Spain, using ARMA- and N4SID-models to predict the energy production, and achieving an accuracy in our models around an 84% in the test dataset. These models have been included in the also developed event-based simulation framework, capable of selecting the best model that achieves a higher accuracy under variable input parameters. The wellness of our approach has been evaluated with data monitored in the production set-up.

As future work, we are planning i) to include more predictive models, including supervised machine learning classification and regression techniques; ii) to adapt the described system to the generation of solar energy; iii) to add both systems in a smart grid scenario that contains different forms of generation and electrical storage, and to optimize the electrical consumption of a high-energy consumer, like a data center.

6 ACKNOWLEDGMENTS

This work has been partially funded by Fundacion Iberdrola España through a Research grant awarded for the 2016-2017 period.

REFERENCES

Ambach, D., and P. Vetter. 2016, Feb. "Wind Speed and Power Forecasting - A Review and Incorporating Asymmetric Loss". In *2016 Second International Symposium on Stochastic Models in Reliability Engineering, Life Science and Operations Management (SMRLO)*, pp. 115–123.

Arroba, P., J. M. Moya, J. L. Ayala, and R. Buyya. 2015, Oct. "DVFS-Aware Consolidation for Energy-Efficient Clouds". In *2015 International Conference on Parallel Architecture and Compilation (PACT)*, pp. 494–495.

Box, G. E. P., G. M. Jenkins, and G. C. Reinsel. 2008. *Time Series Analysis*. John Wiley & Sons, Inc.

Eurostat 2017. "Renewable energy statistics". http://ec.europa.eu/eurostat/statistics-explained/index.php/Renewable_energy_statistics. Accessed November. 23, 2017.

Mao, Y., and W. Shaoshuai. 2016, Oct. "A review of wind power forecasting prediction". In *2016 International Conference on Probabilistic Methods Applied to Power Systems (PMAPS)*, pp. 1–7.

MATLAB 2015. "The MathWorks, Natick, MA, USA". https://es.mathworks.com.

Overschee, P. V., and B. de Moor. 1993. "N4SID: Numerical Algorithms for State Space Subspace System Identification". *IFAC Proceedings Volumes* vol. 26 (2, Part 5), pp. 55 – 58. 12th Triennal Wold Congress of the International Federation of Automatic control. Volume 5 Associated Technologies and Recent Developments, Sydney, Australia, 18-23 July.

Risco-Martín, J. L., S. Mittal, J. C. Fabero, M. Zapater, and R. Hermida. 2017. “Reconsidering the performance of DEVS modeling and simulation environments using the DEVStone benchmark”. *SIMULATION* vol. 93 (6), pp. 459–476.

Sotavento Galicia, S.A. 2017. “Renewable energy statistics”. http://www.sotaventogalicia.com/es/datos-tiempo-real/historicos. Accessed November. 23, 2017.

Tan, K. H., T. Logenthiran, and W. L. Woo. 2016, Nov. “Forecasting of wind energy generation using Self-Organizing Maps and Extreme Learning Machines”. In *2016 IEEE Region 10 Conference (TENCON)*, pp. 451–454.

Tomes, E., and N. Altiparmak. 2017, Sept. “A Comparative Study of HDD and SSD RAIDs’ Impact on Server Energy Consumption”. In *2017 IEEE International Conference on Cluster Computing (CLUSTER)*, pp. 625–626.

Zeigler, B. P., H. Praehofer, and T. G. Kim. 2000. *Theory of Modeling and Simulation. Integrating Discrete Event and Continuous Complex Dynamic Systems*. 2 ed. Academic Press.

AUTHOR BIOGRAPHIES

LAURA PÉREZ-VILARELLE received a degree in Mathematics from University of Santiago de Compostela. She is currently a researcher at Complutense University of Madrid. Her research interests include modeling, simulation and optimization. Her email address is laurap12@ucm.es.

JOSÉ L. RISCO MARTÍN received his Ph.D. from Complutense University of Madrid, and currently is Associate Professor in the Department of Computer Architecture and Automation at Complutense University of Madrid. His research interests include computer aided design, optimization and discrete event simulation. His email address is jlrisco@ucm.es.

JOSÉ L. AYALA received his Ph.D. from Technical University of Madrid in Electrical and Computer Engineering, and currently he is an Associate Professor in the Department of Computer Architecture and Automation at Complutense University of Madrid. His research interests include the development of energy optimization techniques in complex computing scenarios like data centers or smart-cities. His email address is jayala@ucm.es.