Renewable Energy 87 (2016) 892–902

Contents lists available at ScienceDirect

Renewable Energy

journal homepage: www.elsevier.com/locate/renene

Automatic identification of wind turbine models using evolutionary multiobjective optimization

William La Cava [a, *], Kourosh Danai [a], Lee Spector [b], Paul Fleming [c], Alan Wright [c], Matthew Lackner [a]

[a] Dept. of Mechanical and Industrial Engineering, University of Massachusetts Amherst, Amherst, MA 01003, USA
[b] School of Cognitive Science, Hampshire College, Amherst, MA, USA
[c] National Renewable Energy Laboratory, 15013 Denver West Parkway, Golden, CO, USA

ARTICLE INFO

Article history:
Received 8 April 2015
Received in revised form
25 September 2015
Accepted 28 September 2015
Available online 29 October 2015

Keywords:
Wind energy
System identification
Genetic programming
Multiobjective optimization

ABSTRACT

Modern industrial-scale wind turbines are nonlinear systems that operate in turbulent environments. As such, it is difficult to characterize their behavior accurately across a wide range of operating conditions using physically meaningful models. Customarily, the models derived from wind turbine data are in 'black box' format, lacking in both conciseness and intelligibility. To address these deficiencies, we use a recently developed symbolic regression method to identify models of a modern horizontal-axis wind turbine in symbolic form. The method uses evolutionary multiobjective optimization to produce succinct dynamic models from operational data while making minimal assumptions about the physical properties of the system. We compare the models produced by this method to models derived by other methods according to their estimation capacity and evaluate the trade-off between model intelligibility and accuracy. Several succinct models are found that predict wind turbine behavior as well as or better than more complex alternatives derived by other methods. We interpret the new models to show that they often contain intelligible estimates of real process physics.

1. Introduction

As wind energy grows across the globe and new offshore wind turbine installations encounter new operating environments, the models that inform the design and control of these multimillion-dollar machines become increasingly important. Typical multi-megawatt wind turbines exhibit nonlinear behavior and are subject to wind (and sometimes wave) disturbances that are often hard to estimate. These properties make the simulation of their dynamics not only challenging but also site-dependent, because of the influence of wind, wave, and foundation characteristics. Accordingly, the first-principles models of wind turbines, such as the one embedded in the aero-hydro-elastic simulation tool FAST [1], are prone to cumulative discrepancies between prediction and reality. These models are also computationally expensive to run because of their fairly comprehensive representation of wind turbine dynamics. Although the use of engineering models is fundamental to the structural design and loads analysis process, model-based controllers preferably rely on a customized model of the real system in the field, rather than a first-principles model that may miss key elements present in the real system [2].

As an alternative to potentially inaccurate and computationally expensive first-principles models, empirical models of wind turbines are obtained from experimental data to provide a customized representation of the wind turbine. These models are usually in the form of auto-regressive moving-average (ARMAX) models [2–5], neural networks [6], or fuzzy logic models [7], among others, to provide the structural flexibility for adapting the model according to the measured observations. Although these empirical models provide an effective means of estimation/prediction, they have the major drawback of lacking transparency about the physics of the process [8]. This lack of transparency obscures the knowledge of the process that is gained through their development. Ideally, the model should not only be accurate, but intelligible so that the user acquires the insight attained through the model's development. A well-formed model serves two purposes: (i) it improves knowledge of the underlying dynamics of the system; and (ii) it improves the ability of the wind turbine controller to extract power and minimize loads on the turbine.

* Corresponding author.
E-mail address: wlacava@umass.edu (W. La Cava).

http://dx.doi.org/10.1016/j.renene.2015.09.068
0960-1481/

In order to improve the intelligibility of adapted models, empirical models in the form of symbolic equations can be formulated by symbolic regression [9,10]. In symbolic regression, the process variables, inputs, and parameters (constants) are treated as symbols and integrated as blocks to form candidate model structures. Free of restrictions from the form (structure), the search is typically conducted by genetic programming (GP) for candidate models having the best-fit outputs to the measured observations [9]. However, in the absence of a presumed model structure and guided only by the prediction error (i.e., the difference between the modeled and measured outputs), symbolic regression often yields illegible, albeit accurate, models that do not convey any of the physics of the process. The method proposed for modeling here safeguards against this potential shortcoming by two innovations. First, it uses a novel GP method known as epigenetic linear genetic programming (ELGP) that combines the flexibility of stack-based GP representations with an epigenetic encoding to allow for topological search of the candidate model structures, leading to less complex and more accurate results than traditional GP [11,12]. Second, it uses an evolutionary multi-objective optimization (EMO) framework [13] that includes the complexity of the model as an objective in order to yield accurate models that are as intelligible as possible.

In this paper we evaluate the applicability of the proposed ELGP method in identifying wind turbine models based on experimental data collected in normal closed-loop operation from the three-bladed Controls and Advanced Research Turbine (CART3), a turbine maintained by the National Renewable Energy Laboratory (NREL). The paper is organized as follows. First, we present a brief overview of wind turbine mechanics. We then review previous system identification work. Next, the problem formulation as sought by multiobjective optimization is presented, followed by a description of the proposed ELGP method. We then detail the wind turbine identification procedure and analyze results pertaining to local and global models of the wind turbine. The paper concludes with a discussion of the intelligibility of the identified models as they inform the physics of the process.

2. Wind turbine mechanics

Identification of wind turbine models is a difficult undertaking because of the many layers of nonlinearity governing their behavior. Moreover, modern horizontal-axis wind turbines (HAWTs) are controlled using variable-speed and variable-blade pitch operation, further complicating the dynamics. Consider for instance the steady-state aerodynamic rotor torque (Q_R) and thrust (T_R) generated by the rotor operating in freestream wind speed V, defined by:

$$Q_R = \frac{1}{2}\rho\pi R^3 C_q(\lambda,\beta)V^2 \tag{1}$$

$$T_R = \frac{1}{2}\rho\pi R^2 C_T(\lambda,\beta)V^2 \tag{2}$$

where the tip speed ratio $\lambda = \Omega R/V$ relates the rotor speed Ω to the wind speed V, ρ is the air density, R is the rotor radius, β is the pitch angle of the blades (assumed pitching collectively). C_q and C_T are the torque and thrust coefficients, respectively, defining the corresponding generated lift as functions of λ and β. The overall C_q is a function of local aerofoil drag and lift coefficients C_d and C_l, local incidence angle with the wind, ϕ, and local tip speed ratio λ_r, defined by the strip theory calculation of C_q, as:

$$C_q = \left(8/\lambda^3\right)\int_{\lambda_h}^{\lambda} \sin^2\phi(\cos\phi - \lambda_r\sin\phi)(\sin\phi + \lambda_r\cos\phi)\left[1 - \frac{C_d}{C_l}\cot\phi\right]\lambda_r^2 d\lambda_r \tag{3}$$

Because it is difficult to obtain the lift and drag coefficients at each position along the blade due to small inconsistencies in fabrication and local shape deflections, they are often estimated empirically [14]. The inaccuracy of estimated nonlinear coefficient surfaces C_q and C_T, compounded with the measurement uncertainty and stochasticity of V, impedes prediction of the aerodynamic torque and thrust response of the system.

Control actions are limited to actuating the collective pitch β, the generator torque T_G, and the yaw angle ψ. Because of the highly nonlinear nature of the wind turbine behavior, a pitch action of the same magnitude may result in very different aerodynamic forces depending on the instantaneous wind speed and rotor speed, requiring the employment of gain scheduling for pitch control [15]. In addition to aerodynamic nonlinearities, the turbine has low-frequency periodic excitations induced by the rotating blades at once-per-revolution (1P) and thrice-per-revolution (3P) that are normally within the same frequency range as the fore-aft (FA) and side–side (SS) natural frequencies of the tower, requiring the added provision of avoiding dynamic coupling between these excitations and that of the pitch control that affects Ω. Similarly, the first mode of the wind turbine drivetrain can be excited by the generator torque commands, so the generator control must account for this fundamental design objective as well. From the above anecdotes it follows that an accurate model of the wind turbine is essential for designing a reliable controller. This need for model accuracy motivates data-based modeling approaches that can account for turbine-specific observations and provide confident estimates of wind turbine behavior.

3. Related work

Most system identification attempts at modeling wind turbines have focused on producing linear time-invariant (LTI) models via ARMAX models [2,4] or modified forms of closed-loop subspace identification (SSID) [3,5]. Although LTI models seem to be effective in characterizing simulated wind turbine behavior at specific operating wind speeds [2,4], they provide only localized representation. As a remedy, SSID methods have been extended to account for the time-varying, nonlinear dynamics of the wind turbines to form global models. For example, Van der Veen [5] showed that Wiener and Hammerstein systems could be used to identify global wind turbine dynamics by providing the model with the nonlinear aerodynamic torque and thrust relations (Eqs. (1) and (2)), as well as the surface functions for (C_p) and (C_T) that vary with the tip speed ratio λ and pitch angle β. This approach, however, requires good knowledge of these two surface functions, which rely on first principles. Another approach to global modeling associates the nonlinearities with the azimuth angle of the rotor and uses a linear parameter-varying (LPV) model to conduct closed-loop identification of the wind turbine dynamics [3]. In this case, the dynamics of the turbine are assumed to vary periodically, so the matrices of the state space model are defined in terms of the azimuth position of the rotor. This approach provides good predictions of the hub moments at the rotor and tower top motion.

The above approaches, albeit in 'black-box' form, are attractive because of their incorporation of expert knowledge in modeling some of the nonlinearities and for their accommodation of control

design. Ideally, however, the system identification approach has the flexibility to work when the aerodynamic properties of the wind turbine and/or the sources of its nonlinear behavior are not well-characterized. In addition, the methods above used special operating conditions in which the input actions (e.g., β, T_G) are perturbed in a pseudo-random binary fashion to minimize the correlation of output and input noise in closed-loop operation. This approach is problematic because it is not always possible for a control engineer to apply excitation signals to the operating turbine, nor is it straightforward to persistently excite the system adequately [2]. For this reason we focus our identification on normal operating data.

There have been some attempts to construct wind turbine models under similarly reduced sets of assumptions, although most focused solely on power prediction, e.g., from wind measurements [16] or from low resolution supervisory control and data acquisition (SCADA) data [6,17]. Kusiak [17] demonstrated that a neural network model and a controller designed via evolutionary computation could improve simulated power output in below-rated conditions. A drawback of this approach is that the models need to be periodically regenerated to continue to perform well, suggesting an overfitting scenario. The method we propose differs from typical data mining in that it precludes structural assumptions for the model and focuses on the derivation of simple, explicative models that are valuable for their intelligibility in addition to their estimation capacity.

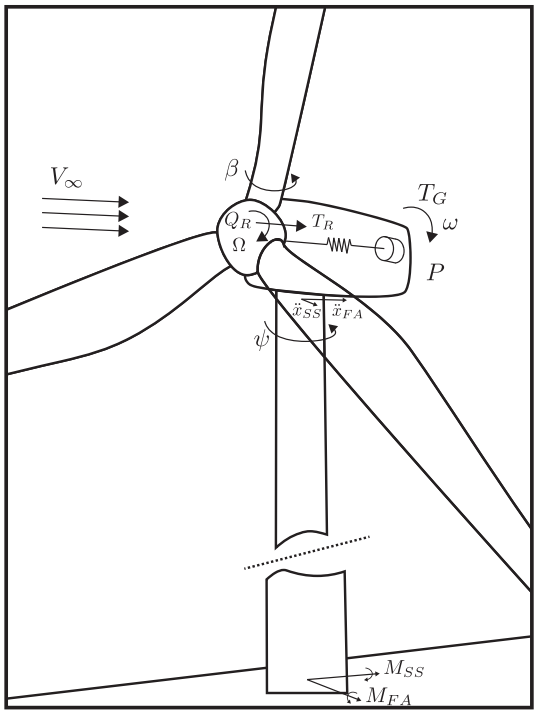

Fig. 1. The wind turbine mechanics considered for identification.

4. Problem statement

The underlying assumption of symbolic regression is that there exists an analytical model of the system that would generate the measured observations $y(t_k)$ at the sample times $t_k = t_1,\ldots,t_N$ under the input, $\mathbf{u}(t)$, as:

$$y(t_k,\mathbf{u}) = \hat{y}(t_k, M^*(\mathbf{x},\mathbf{u},\boldsymbol{\Theta}^*)) + \nu; \quad k = 1,\ldots,N \tag{4}$$

where $\hat{y}$ is the modeled output, ν represents measurement noise in y, $\mathbf{x} = [x_1,\ldots,x_n]^T$ is the vector of state variables, and $M^*(\mathbf{x},\mathbf{u},\boldsymbol{\Theta}^*)$ denotes the correct model form embodied by the correct parameter values $\boldsymbol{\Theta}^*$, written M^* hereafter for brevity. In the search for the correct model form M^*, GP typically attempts to solve the problem:

$$\text{minimize } f(M) \quad \text{subject to } M \in \mathfrak{S} \tag{5}$$

where $\mathfrak{S}$ is the space of possible models M, and f denotes a minimized fitness function. Given that it is impractical to exhaustively search $\mathfrak{S}$, the model found to minimize $f(M)$, denoted $\widehat{M}$, may only be locally optimal, and because M^* is not known, it is possible that $\widehat{M} \neq M^*$. For practical purposes it is assumed that a suboptimal model can adequately represent the process. Typically, a single fitness function quantifies the difference between the target output y and a candidate output $\hat{y}$; however, there is often more than one objective to consider for evaluating the model, in which case the problem becomes:

$$\begin{aligned} &\text{minimize } f_j(M), \quad j = 1,\ldots,J \\ &\text{subject to } M \in \mathfrak{S} \end{aligned} \tag{6}$$

The multiple objective function $\mathbf{f} = [f_1 \ldots f_J]$ would ideally yield a set of nondominated solutions $\tilde{\mathbf{M}} = \{\tilde{M}_1 \ldots \tilde{M}_n\}$, comprising the set of solutions that are Pareto-optimal in $\mathfrak{S}$, where model dominance is defined as:

Definition: Model M_1 dominates M_2; i.e., $(M_1 \prec M_2)$ if $f_j(M_1) \leq f_j(M_2) \forall j$ and $f_j(M_1) < f_j(M_2)$ for at least one j.

In lieu of an exhaustive search of $\mathfrak{S}$, the goal of EMO is to return a set of models $\widehat{\mathbf{M}}$ as close to the Pareto-optimal set $\tilde{\mathbf{M}}$ as possible. It may be easier to represent an arbitrary set of data by a complex model, but it is more difficult to understand and generalize the information content of such a model. Therefore, the solutions from the search must provide a balanced trade-off between accuracy and complexity. Population-based optimization methods like GP are well-suited to address the conflicting objectives of accuracy and conciseness because the solution set $\widehat{\mathbf{M}}$ offers multiple candidate models for approximating $\tilde{\mathbf{M}}$. In the following section we describe a recent symbolic regression method designed to address such a trade-off that is used to conduct the identification of the wind turbine models in this paper.

5. Proposed method

In symbolic regression, the search for candidate models is conducted by GP, whereby a population of computer programs that produce models of the process are evolved. Mathematical building blocks compose the genotype of each program, which is optimized by an evolutionary algorithm. The optimization process, shown in Fig. 2, starts with randomly constructed programs that are repeatedly assessed for their fitness and selectively recombined and mutated until an adequate solution is produced.

5.1. Epigenetic linear genetic programming

In comparison to system identification methods that presume fixed model structures, symbolic regression can be computationally expensive. Furthermore, when guided solely by an error metric, it can yield unwieldy equations that are void of physical meaning. To address these potential shortcomings, we have introduced a symbolic regression method known as ELGP[1] that has been shown to outperform GP on several benchmark regression problems in terms of the conciseness of the developed models, their fitness

[1] Source code available from http://www.github.com/lacava/ellen.

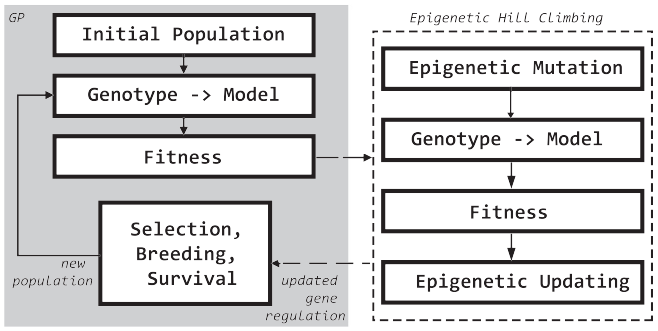

Fig. 2. Block diagram of ELGP. The typical GP steps are shown on the left. After fitness evaluation and before selection, the population undergoes an iteration of epigenetic hill climbing, represented by the block on the right.

convergence, and efficiency of the search [11,12]. This method has two salient features that improve its performance: (i) it uses linear, stack-based programs to represent equations, and (ii) it conducts local search of the space of model structures to both improve the fitness and reduce the complexity of models. We use ELGP in an EMO scheme to maintain and optimize a population of solutions of varying complexity and accuracy.

5.1.1. Model representation

In typical GP systems, equations are represented as parse trees [9] in which the nodes of the tree are mathematical operators (e.g., +, −, *, sin) and the leaves are variables and constants (literals). In contrast, we represent equations by post-fix notation in linear programs that are evaluated using stacks [18,19]. For example, the equation $(x + y)$ can be represented by the program $\mathbf{i} = [x \quad y \quad +]$. The advantage of the proposed stack-based GP system representation is its guarantee of syntactic validity for arbitrary sequences of instructions. It also allows instructions to be silenced or activated in a genotype without invalidating the program's ability to execute. This "syntax-free" property stands in contrast to tree-based representations that can become syntactically invalid due to changes to instructions and literals.

The syntactic robustness of the stack-based approach is achieved mainly by ignoring the execution of instructions that have an arity larger than the current size of the stack. For example, if a + operator attempts to execute and there is only one element on the stack, it does nothing. Furthermore, we base a program's behavior only on the top element(s) of the stack after execution, making genotypes safe with respect to unused arguments. The flexibility of this representation means that the genotypes of the following three programs $\mathbf{i_1}$, $\mathbf{i_2}$, and $\mathbf{i_3}$ all produce the identical model $(x + y)$:

$$\begin{aligned} \mathbf{i}_1 &= [x \quad y \quad +] && \Rightarrow \quad M_1 : (x+y) \\ \mathbf{i}_2 &= [x \quad y \quad + \quad - \quad * \quad /] && \Rightarrow \quad M_2 : (x+y) \\ \mathbf{i}_3 &= [z \quad + \quad x \quad / \quad x \quad y \quad +] && \Rightarrow \quad M_3 : (x+y) \end{aligned} \tag{7}$$

The executions of −, *, and / in $\mathbf{i_2}$ are ignored due to insufficient stack size. In $\mathbf{i_3}$, the last element of the executed stack, $(x + y)$, is taken as the model.

5.1.2. Epigenetic encoding

In GP, the role of epigenetics, that is, regulation of gene expressions, is traditionally ignored, despite the fact that the expression of biological genes is highly regulated and that epigenetic processes provide several evolutionary benefits [20,21]. In contrast, we take advantage of the stack-based representation to introduce epigenetic information by including a boolean condition on each element in an individual's genotype. When evaluated together, the expressed program, i.e., the model, is produced by executing instructions that are **on** (active) and ignoring the instructions that are **off** (inactive). In this light, the genes that were ignored in programs $\mathbf{i}_2$ and $\mathbf{i}_3$ in Eq. (7) provide local solutions to explore in the search space, making it possible to alter the topology and values of the resultant model. For example, program $\mathbf{i_3}$ can admit several models via epigenetic transformations, including:

$$\begin{aligned} \mathbf{i}_3 \rightarrow \mathbf{i}'_3 &= \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ z & + & x & / & x & y & + \end{bmatrix} \Rightarrow M'_3 : (z+y) \\ \mathbf{i}_3 \rightarrow \mathbf{i}''_3 &= \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ z & + & x & / & x & y & + \end{bmatrix} \Rightarrow M''_3 : (z/x) \\ \mathbf{i}_3 \rightarrow \mathbf{i}'''_3 &= \begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 1 & 1 \\ z & + & x & / & x & y & + \end{bmatrix} \Rightarrow M'''_3 : (z/x+y) \end{aligned} \tag{8}$$

Similarly, program $\mathbf{i}_2$ in Eq. (7) admits the models $(x + y)$, $(x - y)$, $(x * y)$, and (x/y) via epigenetic transformations.

5.2. Evolutionary multiobjective optimization

We use three objectives to drive evolutionary pressure during optimization: variance accounted for (VAF), model complexity, and the age of the program in the population. The first two objectives are designed to achieve model accuracy and simplicity. The third objective is used to prevent premature convergence. The three objectives are described in more detail below.

- VAF: We assess the accuracy of each candidate program $\mathbf{i}$ using the VAF metric, which characterizes the normalized variance of the prediction error as:

$$\mathrm{VAF}(\mathbf{i}) = \max\left(0, \frac{1 - \mathrm{var}(y - \hat{y})}{\mathrm{var}(y)}\right) \times 100 \tag{9}$$

Equation (9) is transformed into a minimized objective function as $f_{VAF}(\mathbf{i}) = 1 - VAF(\mathbf{i})/100$.

- Model Complexity: There are several ways to represent the complexity of a model. For example, one can count the number of nodes in the parse tree, or calculate the order of a Chebyshev polynomial fit to the model's output [22]. Here, we account for model complexity by assigning component function nonlinearities to genotype components [23]. Given the following active genotype $\mathbf{g_a} = [g_{a_1} \quad \ldots \quad g_{a_\ell}]$ for program $\mathbf{i}$, the complexity $C(\mathbf{i})$ is defined as:

$$C(\mathbf{i}) = \sum_{q=1}^{\ell} c\left(g_{a_q}\right) \tag{10}$$

with the component function nonlinearities defined as:

$$c(g_a) = \begin{cases} 4 & : \quad (g_a = \log) \vee (g_a = \exp) \\ 3 & : \quad (g_a = \sin) \vee (g_a = \cos) \\ 2 & : \quad (g_a = /) \\ 1 & : \quad \text{otherwise} \end{cases} \tag{11}$$

- Age: Age was originally proposed as a way to layer populations during evolution [24] and later proposed as an objective in a multiobjective scheme [25]. The age of a model in the

population is the number of generations since its oldest ancestor was created. To create age stratification, we introduce a new individual with age 0 to the population each generation. The use of age as an objective protects younger models from being dominated by older ones that are more fit and/or less complex. Furthermore, because younger individuals dominate older ones that may be otherwise equivalent, the introduction of age as an objective pressures the models to improve in fitness and/or complexity with increasing generations, which helps avoid premature convergence.

In our implementation, selection for recombination and mutation is random, meaning all models in the population have the same chance to be selected to produce offspring regardless of their fitness. During this phase, a portion of the population produces children through single point crossover (the recombination of two programs into two new programs) and another portion goes through uniform mutation, in which small changes are made throughout the genome. During recombination and mutation, we create a number of children equal to the overall population size. In addition, a new individual is created each generation as a means of random restart.

The role of the objective functions is to choose the models that survive in the population. Therefore, at the end of each generation, environmental selection is conducted according to the Strength Pareto Evolutionary Algorithm 2 (SPEA2) [26] to reduce the size P of the set consisting of the current population and the newly created individuals down to the original population size N. This algorithm uses two measures to perform this reduction: i) Pareto strength of an individual, $S(\mathbf{i})$, which is the number of individuals equal to or dominated by $\mathbf{i}$, divided by $P+1$, and ii) a density estimate $D(\mathbf{i}) < 1$, based on the inverse of the distance to the k-th nearest neighbor [27] of $\mathbf{i}$ in objective space (in this case the objectives are normalized between zero and one). These metrics are used to define a fitness value $F(\mathbf{i})$ that combines the total strength S of the individuals $\mathbf{j} \in P$ that dominate $\mathbf{i}$ with density estimate $D(\mathbf{i})$:

$$F(\mathbf{i}) = \sum_{j \in P, j < i} S(\mathbf{j}) + D(\mathbf{i}) \tag{12}$$

Every nondominated solution is first copied to the new population. If the new population size is smaller than N, individuals are added in order of lowest $F(\mathbf{i})$. If the population is larger than N, i.e., there are more than N nondominated solutions, individuals are removed iteratively based on $D(\mathbf{i})$. For the latter scenario, the use of $D(\mathbf{i})$ for selection helps preserve the range and spread of solutions along the Pareto front.

It is important to guarantee that all solutions that are succinct and accurate are saved during optimization. With this in mind, an archive is kept updated each generation that contains all nondominated individuals according to only the metrics of VAF and model complexity. This archive provides the solutions that are explored later in this paper.

5.3. Epigenetic learning and evolution

In order to explore local variants of programs by epigenetic adaptation, we implement an epigenetic hill climbing step each generation. During this step, a program's epigenome is uniformly mutated with a probability of 10% at each gene. The mutation flips the binary value of the epiline at the gene, thus activating or silencing that gene. The model resulting from the epigenetic changes $\mathbf{i} \rightarrow \mathbf{i}'$ is then evaluated to determine whether $\mathbf{i}$ should be updated. Lower-complexity programs with equivalent VAF are accepted, giving the following condition:

$$pass = (f_{VAF}(\mathbf{i}') < f_{VAF}(\mathbf{i})) \vee ((f_{VAF}(\mathbf{i}') = f_{VAF}(\mathbf{i})) \wedge (C(\mathbf{i}') < C(\mathbf{i}))) \tag{13}$$

If the epigenetically mutated individual $\mathbf{i}'$ does not pass Eq. (13), the changes are discarded and $\mathbf{i}$ is kept in the population. Otherwise $\mathbf{i}$ is replaced with $\mathbf{i}'$.

6. System identification of CART3

The proposed method is evaluated in application to experimental data from the CART3 system. NREL's CART3 is instrumented with numerous sensors to make the identification of various system models possible. The nature of experimental data available from this system, including its instrumentation, data collection procedure, and control system, is described first. We then describe the settings used for ELGP, including general and problem-specific settings, followed by the types of models considered for identification. We also utilize other system identification approaches to benchmark the ELGP results.

6.1. CART3 system

The CART3 is a 600 kW wind turbine, down-rated to 550 kW, that acts as a test bed for field research at the National Wind Technology Center. It is a three-bladed machine that operates with collective pitch and variable speed control. The CART3 has been instrumented extensively [28] so that data-driven models of the turbine can be established. Among the variables measured are the generator speed ω, rotor speed Ω, pitch action β, generator torque command T_G, tower top acceleration in the fore-aft ($\ddot{x}_{FA}$) and side−side ($\ddot{x}_{SS}$) directions, tower moments in the fore-aft (M_{FA}) and side−side (M_{SS}) directions, and measured power P, as shown in Fig. 1. In addition, an estimate of V is obtained from a meteorological tower located upwind of the turbine. Wind measurements are notoriously uncertain and although methods exist to obtain a better estimate of V [5], they assume good knowledge of C_T and rotor inertia, which are assumed unknowable in our modeling exercise.

To understand the experimental results obtained from the CART3, the controllers used by this system [15,29] are briefly reviewed. The system consists of separate torque and pitch controllers. At wind speeds below rated-power conditions and above cut-in (Region 2 in Fig. 3), the blade pitch β is held constant at an estimated optimum while the generator torque T_G is adjusted proportionately to ω^2 to achieve a theoretical maximum power coefficient C_P. Conversely, at wind speeds above rated-power

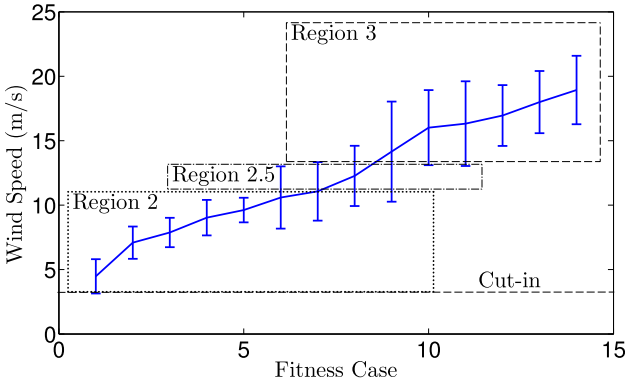

Fig. 3. Mean wind speeds for the 5 min data sets that comprise the training and validation sets. The bottom dotted line indicates the cut-in speed and the regions of turbine operation are marked. Bars indicate the standard deviation of the wind speed.

conditions (Region 3 in Fig. 3), T_G is held constant and the blade pitch β is adjusted by a proportional plus integral (PI) controller to maintain the reference generator speed at the rated power. As such, both closed-loops rely on the accuracy of the model representing the generator speed ω in terms of the corresponding control effort, as the sole input to both systems. In addition to controlled adjustments, β demand is filtered at the tower's first FA and SS modal frequencies to avoid excitation and T_G is filtered to add damping to the drivetrain's first torsional natural frequency. Although the control system is equipped to measure $\ddot{x}_{FA}$ and $\ddot{x}_{SS}$ for damping tower acceleration by adjusting β [15,28], this control strategy was not used during these experiments.

6.2. Identification procedure

For system identification, we used operating data from 14 different 5 min operating periods. The data were collected at 400 Hz and down-sampled to 20 Hz before system identification. The data corresponded to normal operating conditions associated with wind speeds ranging from 2.2 to 25.4 m/s, as shown in Fig. 3, including several start-up and shutdown events. We performed local model identification on each data set separately as well as global identification on the entire combined set (70 min of data). During identification, 30% of the data were chosen randomly from the set to be withheld for validation.

Models were obtained for ω, Ω, M_{FA}, M_{SS}, and P using ELGP with the settings shown in Table 1. We considered three types of models: static models (SMs), in the form of $M(\mathbf{u},\boldsymbol{\Theta})$, first-order discrete-time models (DTMs), in the form of $M(\hat{y}(t_k-1),\mathbf{u}(t_k),\mathbf{u}(t_k-1),\boldsymbol{\Theta})$, and first-order DTMs for one-step look-ahead predictions (DTM-LAs), in the form of $M(y(t_k-1),\mathbf{u}(t_k),\mathbf{u}(t_k-1),\boldsymbol{\Theta})$. The main difference between DTM and DTM-LA forms is their reliance on past predictions versus experimental data. As such, DTMs evaluate the effectiveness of ELGP in a simulation-based environment, in which the output is generated entirely according to the past estimated outputs, whereas DTM-LAs evaluate the scenario of the outputs estimated according to the past values of the measured outputs. The constant values $\boldsymbol{\Theta}$ were initialized as ephemeral random constants [9] picked uniform-randomly from the range $[-10, 10]$. They were then optimized across the population each generation using a stochastic hill climbing algorithm [30]. The hill climber perturbed all constant values in the active genotype by Gaussian noise with a standard deviation equal to 10% of the value of the constant. These changes were kept if they resulted in a lower f_{VAF} for the individual. To prevent the search from focusing on constant optimization, and considering the insensitivity of the fitness metric f_{VAF} to linear transformations of the model output, the variables and outputs were scaled prior to adaptation, for example by rated torque, rated power, or cut-out wind speed.

A key advantage of symbolic regression is *feature selection*: the ability to select the variables to be considered in the model. We expect this feature to be instrumental in delivering parsimonious model forms that more clearly relate process inputs to model outputs as compared to the traditional system identification methods that are void of this capacity. In this regard, the models produced for global identification were compared to several model types obtained in the form of multiple regression, 20th-order autoregressive exogenous (ARX) models, and nonlinear ARX neural networks (NARX-NN). The NARX-NN contained 10 hidden layers and the weights were trained using back-propagation with the Levenberg–Marquardt algorithm.

7. Results

The performance of the identified local models by ELGP is summarized in Table 2. The results correspond to the final model with the maximum VAF from training. The results indicate that the accuracy of the models of Ω, ω, and P is excellent in all cases except for $\overline{V} = 18.0$ m/s, where the response of the system was flat (i.e., $var(\Omega) < 5e{-}4$), obscuring the dynamics. The DTM model has a better performance for this case. The other two models, M_{FA} and M_{SS}, produce generally accurate outputs, but not as accurate as the other three models. For illustration purposes, the outputs of these two models are compared to their counterparts in the test data in Figs. 4 and 5 in both time and frequency domains. The results indicate that the peak frequencies are captured by the model, including the first tower FA bending mode at ≈ 0.88 Hz. As to model formulations, the DTM forms are slightly more accurate than the SM form for Ω, ω, and P but are similar for the other cases. The overall accuracy of SM and DTM is significantly different only in representing P on training data, according to a Wilcoxon rank sum test ($p = 0.041$).

The performance of global models is summarized in Table 3, where their accuracy is compared with other models commonly reported in the literature. Specifically, the accuracy of SM and DTM forms is compared to that of models in the form of multiple regression and 20^{th} order ARX models. The accuracy of the DTM-LA form is compared to that of a NARX-NN. The results in Table 3 indicate excellent accuracy of the models of Ω, ω, and P obtained by ELGP, and the lower accuracy of the models of M_{FA} and M_{SS}, as was also observed with the local models in Table 2. As with the local models, a comparison of the outputs of these models to data in both time and frequency domains, shown in Figs. 6 and 7, indicates accurate representation of low-order dynamics and the peak frequencies. The results in Table 2 also indicate the better accuracy of the SM and DTM models generated by ELGP than that of the linear regression or the ARX model forms. The accuracy difference is particularly pronounced in the M_{FA} and M_{SS} models as represented

Table 1
Symbolic regression settings.

Setting	Value
Population size	2400
Crossover/mutation	80/20%
Program length limits	[10, 100]
Ephemeral random constant range	[−10, 10]
Termination criterion	2e12 (local)/1e13 (global) gene evaluations
Function set	{+, −, *, /, sin, cos, exp, log}
Output	Dependent variables
Ω	$\{V, \beta, T_G, t\}$
ω	$\{V, \beta, T_G, t\}$
M_{FA}	$\{V, \beta, T_G, t, \lambda, \ddot{x}_{FA}, \ddot{x}_{SS}\}$
M_{SS}	$\{V, \beta, T_G, t, \lambda, \ddot{x}_{FA}, \ddot{x}_{SS}, \psi\}$
P	$\{V, \beta, T_G, \omega\}$

Table 2
Performance of local models generated by ELGP using SM and DTM model formulations. Results are categorized by the mean wind speed ($\overline{V}$) of the corresponding 5 min data set.

Training/validation VAF (%)					
$\overline{V}$(m/s)	Ω	ω	M_{FA}	M_{SS}	P
SM					
4.5 m/s	99.3/99.2	99.4/99.4	97.5/97.3	95.1/94.5	99.8/99.7
7.1 m/s	97.0/96.9	98.0/97.9	80.4/79.1	86.6/85.4	99.9/99.9
7.9 m/s	99.9/99.9	99.8/99.8	80.2/79.1	85.8/85.6	100.0/100.0
8.9 m/s	99.9/99.9	99.8/99.8	90.5/90.3	89.9/89.3	100.0/100.0
9.5 m/s	99.6/99.6	99.3/99.3	83.8/83.7	81.9/81.1	99.8/99.8
10.6 m/s	99.8/99.9	99.7/99.7	91.3/90.9	87.8/87.7	99.8/99.8
10.9 m/s	99.7/99.7	99.7/99.6	92.0/92.0	82.7/82.7	99.8/99.8
12.3 m/s	98.6/98.5	98.1/98.0	79.0/78.9	73.2/71.8	99.4/99.4
14.2 m/s	99.9/99.8	99.8/99.8	95.9/95.9	73.2/73.4	99.8/99.8
16.0 m/s	99.1/99.1	99.0/99.1	94.2/94.2	85.0/83.5	99.9/99.9
16.3 m/s	97.8/97.8	97.1/97.0	71.8/71.0	82.7/82.6	99.6/99.5
17.0 m/s	99.6/99.6	99.5/99.5	83.3/83.0	83.8/83.6	99.9/99.9
18.0 m/s	18.2/17.0	21.8/22.3	52.7/51.0	70.2/70.8	72.3/72.4
18.9 m/s	99.9/99.9	99.9/99.9	92.3/92.7	96.6/97.0	100.0/100.0
DTM					
4.5 m/s	99.4/99.4	99.4/99.3	97.6/97.4	95.3/95.3	99.8/99.8
7.1 m/s	99.2/99.2	99.4/99.3	80.8/80.9	93.2/92.1	99.9/99.9
7.9 m/s	100.0/100.0	100.0/100.0	88.2/89.1	92.3/92.0	100.0/100.0
8.9 m/s	100.0/100.0	100.0/100.0	97.1/96.9	95.9/96.0	100.0/100.0
9.5 m/s	99.8/99.8	99.7/99.7	91.3/90.2	85.6/85.1	99.9/99.9
10.6 m/s	99.9/99.8	99.8/99.7	92.3/92.7	90.4/89.5	100.0/100.0
10.9 m/s	99.8/99.8	99.6/99.6	91.7/90.8	87.0/86.8	100.0/100.0
12.3 m/s	98.8/98.8	98.3/98.3	80.8/79.7	81.4/81.0	99.8/99.8
14.2 m/s	99.8/99.8	99.9/99.9	95.9/96.0	76.7/76.5	99.9/99.9
16.0 m/s	99.1/99.0	99.1/99.0	94.2/93.8	86.5/86.6	100.0/100.0
16.3 m/s	97.9/98.1	97.1/97.6	73.4/74.3	86.1/83.6	99.9/99.9
17.0 m/s	99.5/99.5	99.5/99.5	85.3/85.8	86.2/86.5	100.0/99.8
18.0 m/s	90.0/63.8	81.5/56.1	61.1/60.1	78.5/77.4	76.4/78.1
18.9 m/s	100.0/99.9	99.9/99.9	91.9/91.1	97.3/96.8	100.0/100.0

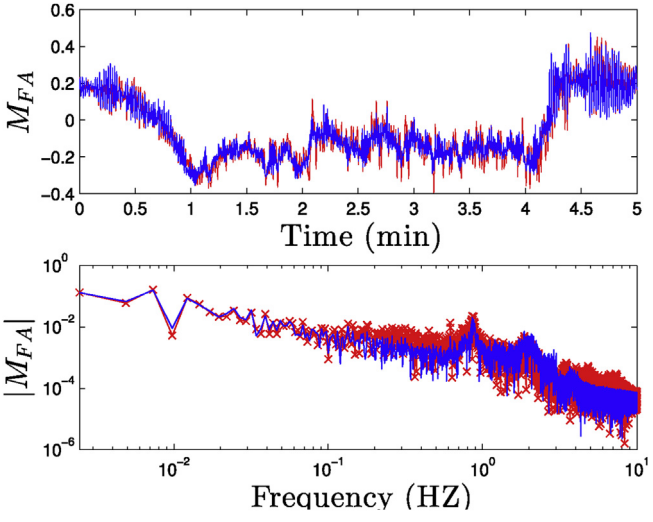

Fig. 4. SM M_{FA} model (blue) with training and validation data (red) at $\overline{V} = 16.0$ m/s. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

by the 45–50% higher VAF values and the models of Ω and ω as characterized by the higher VAF values of approximately 10–28%. As to the performance of the DTM-LA form, which is compared to that of a NARX-NN, the tower moment predictions, M_{FA} and M_{SS}, are significantly better than those of SM and DTM by both methods. Although ELGP and NARX-NN show nearly identical prediction capability in all cases, they differ in transparency (intelligibility), as discussed next.

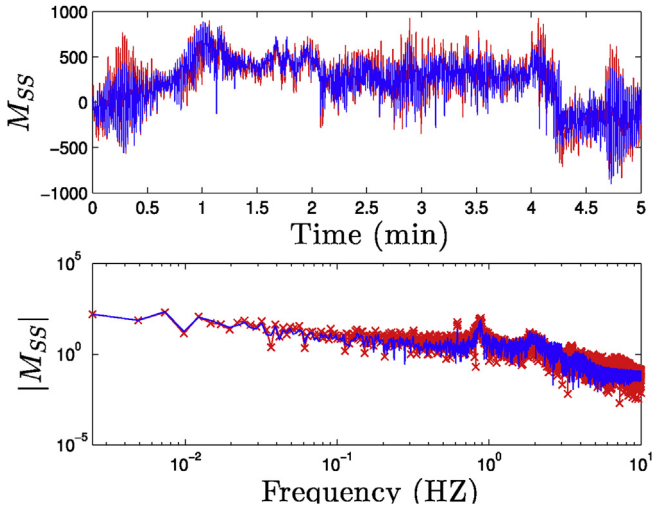

Fig. 5. SM M_{SS} model (blue) with training and validation data (red) at $\overline{V} = 16.0$ m/s. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

7.1. Model interpretation

ELGP maintains an archive of solutions that are nondominated with respect to the objectives of fitness and complexity during identification for the purpose of providing less complex and possibly more general alternatives to the best training solutions. Previous research has suggested that models with physical insight normally reside along the edges of the Pareto set, where a small increase in complexity could result in large improvement in estimation accuracy [31]. We observe this phenomenon in our results, as shown in several of the archives in Figs. 8–11, with "n" being a placeholder for all constants.

7.1.1. Local models

We find that the low complexity models in the archives often identify basic relations in the closed-loop system. For example, consider the local models created by ELGP as illustrated in Figs. 8–10, which show the models on the Pareto front of the archives. The first figure, Fig. 8, corresponds to the models of ω in SM form in below-rated operating conditions where the torque control strategy is $T_G = k\omega^2$. Low-complexity solutions include $\omega = \frac{n}{T_G}$, which is the analytical solution to the power law $P{=}\omega T_G$ when P is constant, and $\omega = (n - T_G)T_G = nT_G - T_G^2$, which in the absence of square root or exponent operators, bears some resemblance to the Taylor series approximation for square root: as $\sqrt{1+x} \approx 1 + \frac{1}{2}x - \frac{1}{8}x^2$, characterizing the relationship $\omega = n\sqrt{T_G}$. The basic closed-loop relationship between ω and β is also identified in DTM form in above-rated operation, as shown in Fig. 9; in this case, a low-complexity model on the edge of the Pareto set is $\omega_k = n(\omega_{k-1} + (\beta - \beta_{k-1}))$, which describes the proportional control effort of β with respect to ω and generalizes better than more complex models. The archives of local power models also contain process physics, as shown in Fig. 10. The relation $P = \omega T_G$ occupies the elbow of the curve, with slight variations of this increasing model accuracy at the cost of complexity. An interesting solution scales the power law by a nonlinear function of the wind disturbance: $P{=}\omega T_G/\sin(e^{V/n})$.

We are able to draw two main insights from the local model archives. First, the feature selection property of GP results in local models that describe the local closed-loop system without inactive control input variables. This is evident in the comparison of the

Table 3
Comparison of global models generated by ELGP (SM, DTM, DTM-LA), two linear system identification methods (multiple regression, ARX), and a neural network (NARX-NN). The one-step prediction models (DTM-LA and NARX-NN) are grouped on the right. The best method for each case is in bold.

	Training/Validation VAF (%)					
	SM	DTM	Multiple regression	20th-order ARX	DTM-LA	NARX-NN
Ω	98.4/96.9	**98.7**/**98.7**	91.9/91.9	71.0/71.0	**100.0**/**99.9**	99.9/99.8
ω	97.8/98.4	**98.6**/**98.6**	92.0/91.9	69.0/69.0	**100.0**/**99.9**	99.9/99.8
M_{FA}	**76.0**/**76.1**	74.2/74.4	31.5/32.2	25.6/25.6	**98.7**/**94.9**	98.6/**94.9**
M_{SS}	69.5/69.6	**72.7**/**72.2**	19.6/20.4	0.0/0.0	**97.6**/89.9	97.3/**90.6**
P	**99.9**/**99.9**	**99.9**/**99.9**	99.7/99.7	99.6/99.6	–/–	–/–

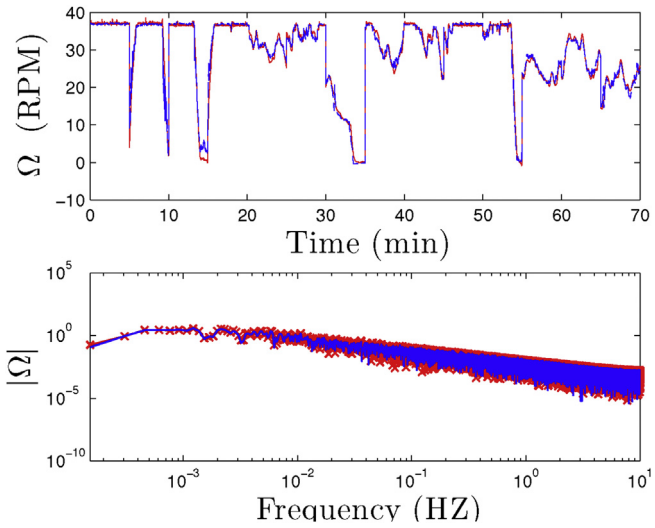

Fig. 6. Comparison of the global Ω SM model (blue) and combined training and validation data (red). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

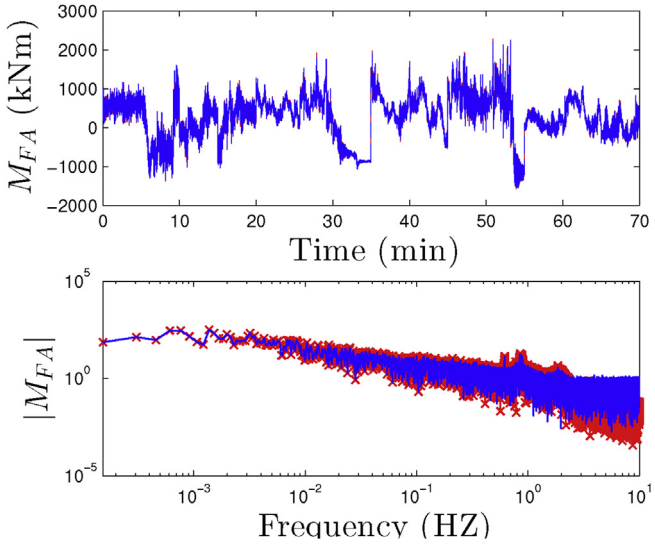

Fig. 7. Comparison of the global DTM-LA M_{FA} model (blue) and combined training and validation data (red). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

below-rated models of ω (Fig. 8) that depend mostly on T_G and the above-rated models (Fig. 9) that depend mostly on β. Second, the appearance of the control strategies in the low-complexity models of the archive indicates that the closed-loop dynamics exhibited by

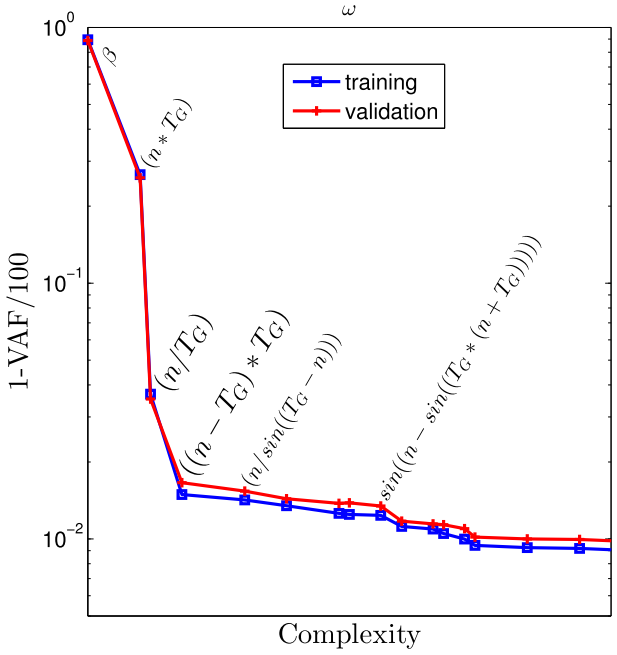

Fig. 8. Pareto archive of SM models of ω at $\overline{V} = 9.5$ m/s.

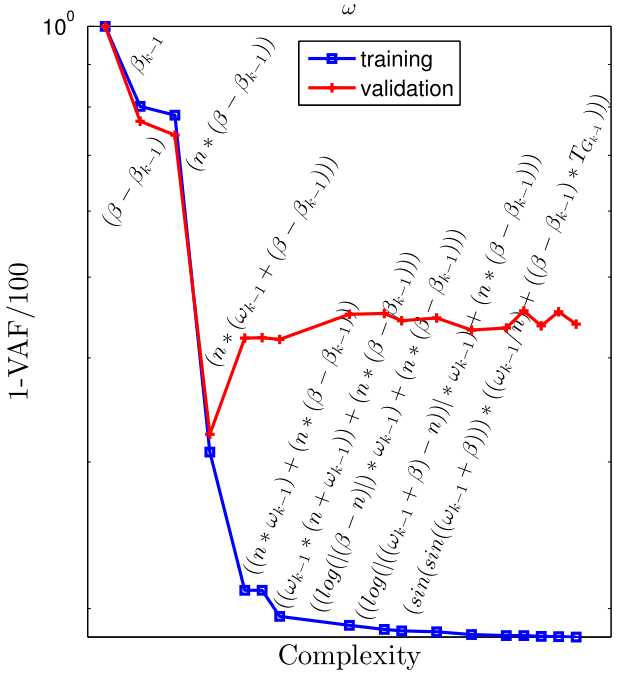

Fig. 9. Pareto archive of DTM models of ω at $\overline{V} = 18.0$ m/s.

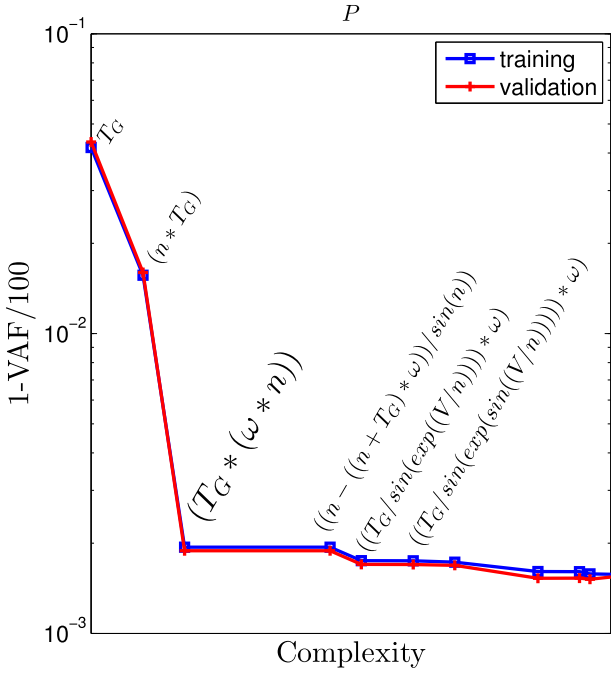

Fig. 10. SM Pareto archive of P at $\overline{V} = 7.1$ m/s.

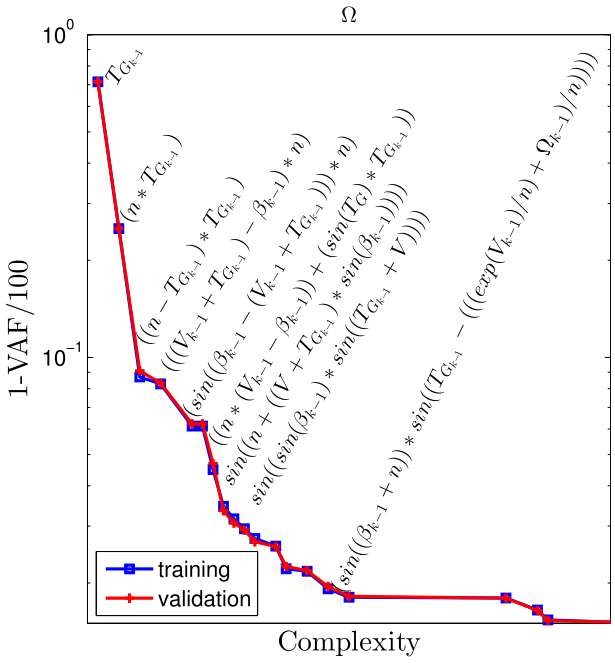

Fig. 11. DTM Pareto archive of global models of Ω.

the wind turbine are in some cases heavily defined by the controller behavior.

7.1.2. Global models

Performing the model search globally has the advantage of testing candidate models across control regimes that vary in their relation to the turbine behavior; this may help distinguish the behavior of the wind turbine from the changing relations governing the inputs. Indeed, we find that the global DTM and DTM-LA models are generally more dependent on several inputs, including their previous outputs and V, which indicates that the behavior of the plant is more uniquely identified. For example, Fig. 11 shows the Pareto archive of Ω from global identification. The final model shown is a nonlinear function of V_{k-1}, β_{k-1}, $T_{G_{k-1}}$, and Ω_{k-1}. It contains the same term $\sin(e^{V/n})$ found in the local power models, suggesting that this may be a concise way for models to represent the nonlinear response of outputs to V.

The global identification of P converges within eight generations on the power law, $P = \omega T_G$, which remains the most accurate global model found of any complexity. Interestingly, the success of the linear predictions of P in Table 3 can be understood using the archive of global P solutions that contains only two nondominated solutions: $P = nT_G$ and $P = nT_G\omega_n$. Hence, the ELGP archive contains a linear model of power based on T_G that has approximately the same VAF value as the linear models (99.2%). Note that the ELGP models are more parsimonious than those generated by the linear methods due to feature selection.

The results in Table 3 show that the ELGP method is able to produce one-step look-ahead models that are as accurate as those produced by a black-box neural network model. Compared to the NARX-NN models that are undecipherable, those generated by ELGP are quite succinct:

$$\tilde{\Omega}_k = \tilde{\Omega}_{k-1} - \sin\left(\frac{n_1}{t}\right)\sin\left(\frac{n_2 \tilde{V}_{k-1}\tilde{T}_{G_{k-1}}}{\tilde{\Omega}_{k-1}(\beta_{k-1} + n_3)}\right) \tag{14}$$

$$\tilde{M}_{FA_k} = \tilde{M}_{FA_{k-1}} + n_1 \sin\left(\tilde{T}_G\right)\left(\ddot{\tilde{x}}_{FA} - \ddot{\tilde{x}}_{FA_{k-1}}\right)/\tilde{V} \tag{15}$$

$$\tilde{M}_{SS_k} = \tilde{M}_{SS_{k-1}} + n_1 \sin(n_2\psi)\left(\ddot{\tilde{x}}_{FA} - \ddot{\tilde{x}}_{FA_{k-1}}\right) \tag{16}$$

Note that $(\tilde{\cdot})$ denotes the scaled variables. In each case the model consists of the summation of the previous output and a compact nonlinear function. For the tower moment cases, the nonlinear component contains the change in fore-aft acceleration of the tower top ($\ddot{\tilde{x}}_{FA} - \ddot{\tilde{x}}_{FA_{k-1}} = \Delta\ddot{\tilde{x}}_{FA}$), which is an intuitive result. Interestingly, the model for M_{SS} decomposes $\Delta\ddot{\tilde{x}}_{FA}$ into its side–side component using the yaw angle, i.e., $\sin(n_3\psi)$, rather than using the measurement of $\ddot{\tilde{x}}_{SS}$. This may indicate that $\ddot{\tilde{x}}_{FA}$ is more reliably measured than $\ddot{\tilde{x}}_{SS}$.

8. Discussion

The local and global models obtained by ELGP using EMO demonstrate the potential for the succinct resultant models to enhance the understanding of the characteristics observed from a process. We have shown that the models are transparent enough to link process estimation to understandable model components (Eqns. 14–16) and accurate enough to shed light on the characteristics of the closed-loop behavior of the system. Furthermore, by studying the archive we are able to see how specific increases in complexity affect the model's estimation capacity. In general, the models do not suffer from overfitting (with some exceptions, e.g., Fig. 9 and the DTM-LA tower moment models). This may be due to the way in which the validation set is chosen or be a property of the model forms that are discovered.

The produced models hold a utility beyond process insight. There is clearly a need for high-fidelity local models of wind turbines for control design [32]. The models can be used directly in a model predictive control system, or linear models of the wind turbine at different wind speeds can be derived if the control design process requires them. These models could be generated from the global nonlinear models by several approaches. One approach is to linearize the model forms analytically; another is to build auto-

regressive models from the output of the nonlinear model in quasi-linear operating regimes. Deriving the linear models directly from the predicted output of the model may provide more flexibility in choosing the structure of the linear model. Even the static models may be utilized as a target for the transformation to a set of local linear dynamic models.

A direct comparison to previous CART3 identification results is difficult given the difference in operating conditions and assumptions, although the results here compare well to those previously published [33] that use perturbance injection. It is likely that the results could be further improved if persistent excitation signals were used or the model formulations were higher than first order. However, we have shown that accurate models can be obtained even with the limited experimental setup we employ. We have chosen to use these restrictions to make the results relevant to data collected from wind turbines during regular operation, which is far more common, and requires less expertise to obtain. The limited assumptions regarding the aerodynamic properties of the turbine broadens the applicability of the work to poorly characterized turbines as well. In terms of computation time, this approach is more efficient than standard GP but incurs a higher cost than linear methods. Typical runs over the global data for DTM model training require several hours to converge. Fitness estimation methods [34] can assist in scaling as the dimensionality of the data set increases.

9. Conclusion

In this work we use a novel symbolic regression system in an evolutionary multiobjective optimization framework to identify compact models of a wind turbine from operating data with minor assumptions. The models are not only accurate, but succinct and intuitive, and have been shown to embody process knowledge in several instances. This method of system identification may be a promising middle ground between conducting computationally expensive physics simulations and using black-box models since the models evaluate quickly while still capturing the intelligible system behavior. In the future we plan to fully characterize how the sets of assumptions about the aerodynamic properties of the turbine and the operating conditions change the fidelity of the identified models.

As wind turbines continue to grow in size and flexibility and begin to move offshore onto floating platforms, we expect data-based modeling of wind turbine behavior to become an even more integral part of the design and research processes. The use of intelligible modeling methods can help catalyze expert knowledge of turbine behavior in response to combined wind and wave loading. Even small gains in power capture of utility-scale wind farms can result in large financial gains, and therefore it is crucial to improve the power capture as well as the lifetime and maintenance of these machines. For this reason, we expect that data-based approaches such as this one are key to continuing the success of wind energy technology worldwide.

Acknowledgments

The authors would like to thank Dr. van der Veen for sharing his insights into identification of wind systems. This work is partially supported by the NSF-sponsored IGERT: Offshore Wind Energy Engineering, Environmental Science, and Policy (Grant Number 1068864), as well as Grant Nos. 1017817, 1129139, and 1331283. This work was also supported by the U.S. Department of Energy (DOE) under Contract No. DE-AC36-08GO28308 with the National Renewable Energy Laboratory. Funding for the work was provided by the DOE Office of Energy Efficiency and Renewable Energy, Wind and Water Power Technologies Office. This work used the Extreme Science and Engineering Discovery Environment (XSEDE), which is supported by National Science Foundation grant number ACI-1053575 [35].

References

[1] Jason M. Jonkman, Marshall L. Buhl Jr., FAST Users Guide, Technical Report NREL/TP-500-38230, 2005.
[2] M. Iribas, I.-D. Landau, Identification of wind turbines in closed-loop operation in the presence of three-dimensional turbulence wind speed: torque demand to measured generator speed loop, Wind Energy 12 (7) (October 2009) 660–675.
[3] J.W. Van Wingerden, I. Houtzager, F. Felici, M. Verhaegen, Closed-loop identification of the time-varying dynamics of variable-speed wind turbines, Int. J. Robust Nonlinear Control 19 (1) (2009) 4–21.
[4] M. Iribas-Latour, I.-D. Landau, Identification in closed-loop operation of models for collective pitch robust controller design, Wind Energy 16 (3) (April 2013) 383–399.
[5] Gijs van der Veen, Jan-Willem van Wingerden, Michel Verhaegen, Global identification of wind turbines using a Hammerstein identification method, IEEE Trans. Control Syst. Technol. 21 (4) (July 2013) 1471–1478.
[6] Andrew Kusiak, Wenyan Li, Zhe Song, Dynamic control of wind turbines, Renew. Energy 35 (2) (February 2010) 456–463.
[7] S. Bououden, M. Chadli, S. Filali, A. El Hajjaji, Fuzzy model based multivariable predictive control of a variable speed wind turbine: LMI approach, Renew. Energy 37 (1) (January 2012) 434–439.
[8] Stephen A. Billings, Nonlinear System Identification: NARMAX Methods in the Time, Frequency, and Spatio-temporal Domains, John Wiley & Sons, 2013.
[9] John R. Koza, Genetic Programming: on the Programming of Computers by Means of Natural Selection, MIT Press, Cambridge, MA, USA, 1992.
[10] Gary J. Gray, David J. Murray-Smith, Yun Li, Ken C. Sharman, Thomas Weinbrenner, Nonlinear model structure identification using genetic programming, Control Eng. Pract. 6 (11) (1998) 1341–1352.
[11] William La Cava, Lee Spector, Kourosh Danai, Matthew Lackner, Evolving differential equations with developmental linear genetic programming and epigenetic hill climbing, in: Companion Proceedings of the 2014 Conference on Genetic and Evolutionary Computation (GECCO), ACM Press, 2014, pp. 141–142.
[12] William La Cava, Thomas Helmuth, Lee Spector, Kourosh Danai, Genetic programming with epigenetic local search, in: Proceedings of the Genetic and Evolutionary Comptutation Conference (GECCO), ACM Press, 2015, pp. 1055–1062.
[13] Kalyanmoy Deb, Multi-objective Optimization Using Evolutionary Algorithms, John Wiley & Sons, July 2001.
[14] J.F. Manwell, J.G. McGowan, Anthony L. Rogers, Wind Energy Explained: Theory, Design and Application, Wiley, Chichester, U.K, 2009.
[15] E.A. Bossanyi, Wind turbine control for load reduction, Wind Energy 3 (3) (2000) 149–163.
[16] Mona N. Eskander, Neural network controller for a permanent magnet generator applied in a wind energy conversion system, Renew. Energy 26 (3) (July 2002) 463–477.
[17] Andrew Kusiak, Haiyang Zheng, Zhe Song, Power optimization of wind turbines with data mining and evolutionary computation, Renew. Energy 35 (3) (March 2010) 695–702.
[18] Timothy Perkis, Stack-based genetic programming, in: Evolutionary Computation, 1994. IEEE World Congress on Computational Intelligence., Proceedings of the First IEEE Conference on, IEEE, 1994, pp. 148–153.
[19] Lee Spector, Alan Robinson, Genetic programming and autoconstructive evolution with the push programming language, Genet. Program. Evolvable Mach. 3 (1) (2002) 7–40.
[20] Eva Jablonka, Marion J. Lamb, The changing concept of epigenetics, Ann. N. Y. Acad. Sci. 981 (1) (2002) 82–96.
[21] Robin Holliday, Epigenetics: a historical overview, Epigenetics 1 (2) (2006) 76–80.
[22] E.J. Vladislavleva, G.F. Smits, D. den Hertog, Order of nonlinearity as a complexity measure for models generated by symbolic regression via pareto genetic programming, IEEE Trans. Evol. Comput. 13 (2) (2009) 333–349.
[23] Guido F. Smits, Mark Kotanchek, Pareto-front exploitation in symbolic regression, in: Genetic Programming Theory and Practice II, Springer, 2005, pp. 283–299.
[24] Gregory S. Hornby, ALPS: the age-layered population structure for reducing the problem of premature convergence, in: Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation, GECCO'06, ACM, New York, NY, USA, 2006, pp. 815–822.
[25] Michael Schmidt, Hod Lipson, Age-fitness pareto optimization, in: Genetic Programming Theory and Practice VIII, Springer, 2011, pp. 129–146.
[26] Eckart Zitzler, Marco Laumanns, Lothar Thiele, SPEA2: Improving the Strength Pareto Evolutionary Algorithm, Eidgenössische Technische Hochschule Zürich (ETH), Institut für Technische Informatik und Kommunikationsnetze (TIK), 2001.
[27] Bernard W. Silverman, Density Estimation for Statistics and Data Analysis, vol. 26, CRC Press, 1986.
[28] Paul Fleming, Jan-Willem Van Wingerden, Alan D. Wright, Comparing State-

Space Multivariable Controls to Multi-SISO Controls for Load Reduction of Drivetrain-Coupled Modes on Wind Turbines Through Field-Testing: Preprint, Technical Report NREL/CP-5000–53500, 2011.
[29] Alan Duane Wright, Modern Control Design for Flexible Wind Turbines, Technical Report NREL/TP-500–35816, 2004.
[30] Josh Bongard, Hod Lipson, Automated reverse engineering of nonlinear dynamical systems, Proc. Natl. Acad. Sci. 104 (24) (2007) 9943–9948.
[31] Michael Schmidt, Hod Lipson, Distilling free-form natural laws from experimental data, Science 324 (5923) (2009) 81–85.
[32] F.D. Bianchi, R.S. Sánchez-Peña, M. Guadayol, Gain scheduled control based on high fidelity local wind turbine models, Renew. Energy 37 (1) (January 2012) 233–240.
[33] Gijs van der Veen, Identification of Wind Energy Systems, 2013.
[34] M.D. Schmidt, H. Lipson, Coevolution of fitness predictors, IEEE Trans. Evol. Comput. 12 (6) (December 2008) 736–749.
[35] John Towns, Timothy Cockerill, Maytal Dahan, Ian Foster, Kelly Gaither, Andrew Grimshaw, Victor Hazlewood, Scott Lathrop, Dave Lifka, Gregory D. Peterson, Ralph Roskies, J. Ray Scott, Nancy Wilkens-Diehr, XSEDE: accelerating scientific discovery, Comput. Sci. Eng. 16 (5) (2014) 62–74.