Equation Discovery for Nonlinear System Identification

NIKOLA SIMIDJIEVSKI[1,2]**, LJUPČO TODOROVSKI**[1,3]**,**
JUŠ KOCIJAN[1,4]**, (Senior Member, IEEE), AND SAŠO DŽEROSKI**[1,5]
[1]Jožef Stefan Institute, 1000 Ljubljana, Slovenia
[2]Department of Computer Science and Technology, University of Cambridge, Cambridge CB3 0FD, U.K.
[3]Faculty of Public Administration, University of Ljubljana, 1000 Ljubljana, Slovenia
[4]School of Engineering and Management, University of Nova Gorica, 5000 Nova Gorica, Slovenia
[5]Jožef Stefan International Postgraduate School, 1000 Ljubljana, Slovenia

Corresponding authors: Nikola Simidjievski (nikola.simidjievski@ijs.si) and Ljupčo Todorovski (ljupco.todorovski@fu.uni-lj.si)

ABSTRACT Equation discovery methods enable modelers to combine domain-specific knowledge and system identification to construct models most suitable for a selected modeling task. The method described and evaluated in this paper can be used as a nonlinear system identification method for gray-box modeling. It consists of two interlaced parts of modeling that are computer-aided. The first performs computer-aided identification of a model structure composed of elements selected from user-specified domain-specific modeling knowledge, while the second part performs parameter estimation. In this paper, recent developments of the equation discovery method called process-based modeling, suited for nonlinear system identification, are elaborated and illustrated in two continuous-time case studies. The first case study illustrates the use of the process-based modeling on synthetic data while the second case-study evaluates process-based modeling on measured data for a standard system-identification benchmark. The experimental results clearly demonstrate the ability of process-based modeling to reconstruct both model structure and parameters from measured data.

INDEX TERMS Machine learning, nonlinear system identification, equation discovery, process-based modeling, computational scientific discovery, knowledge-based identification.

I. INTRODUCTION

Data-driven system identification [19] is omnipresent in many scientific and engineering domains: Given time-series data describing observed system behavior, the task is to find a model, i.e., identify an appropriate model structure and estimate the values of the model parameters. Identifying the model structure is a challenging problem in many practical applications, especially in domains where structure inference from first principles is not an option. In such cases, one can opt for the black-box or the gray-box modeling paradigm.

In black-box modeling, we conjecture that the appropriate model structure belongs to a general class of structures, such as neural networks and fuzzy models [16] or regression trees [1]. While nonlinear black-box models can achieve highly accurate reconstruction of the observed

The associate editor coordinating the review of this manuscript and approving it for publication was Xianye Ben.

system behavior, they do not reveal the structure of the observed system. Namely, there is no clear correspondence between the assumed model structure (e.g., neural network) and the structure of the system at hand.

Gray-box models, on the other hand, aim primarily at revealing the system structure. They are usually formalized as equations that scientists and engineers can comprehend and relate to [20]. To construct gray-box models, methods for symbolic regression [15], [24] and equation discovery [32] are commonly used. Symbolic regression addresses the gray-box modeling task with evolutionary methods, i.e., genetic algorithms [24]. Following these, equation structures are evolved from an initial set of variables, operators, functions and constants. Users have little or no control over the space of equation structures considered during search. Thus, the obtained models often include too complex and incomprehensible equations that make little contact with the modeling knowledge in the domain of use. Hence, the mapping between

the structure of the model equations and the system structure is non-trivial and has to be inferred by human modelers.

Equation discovery methods [18], on the other hand, allow the user to employ domain-specific knowledge and specify the appropriate space of candidate models. To this end, different formalisms for specifying equation fragments as components for building mathematical models have been proposed [9]. While these formalisms can, in principle, be used in the context of evolutionary methods [22], symbolic regression approaches rarely use them. Instead, equation discovery approaches often rely upon exhaustive or greedy search methods [31]. The most recent equation discovery approach, process-based modeling (PBM), is based on a knowledge representation formalism that establishes an explicit mapping between the model structure and the structure of the observed system [5], [32]. It is closely related to component-based approaches for computer-aided design of electronic circuits [2], where models are composed from a library of standardized, well-characterized circuit components. These components have fixed constant parameters, used for manual model construction, where the modeler establishes a configuration of model components. PBM employs a library of similar components in the context of computer-aided input-output modeling, where the selection of an appropriate configuration of components (and the values of their parameters) is based on output-error fit against measured data.

Process-based modeling has been applied in several practically relevant case studies. However, so far, it has only been evaluated partially in a limited setting and has not yet been systematically evaluated on a proper system identification benchmark. On one hand, PBM has been used to establish new models of aquatic ecosystems from data and domain-specific knowledge, but with no reference to other known or well-established models [3]. On the other hand, PBM has been used to reconstruct the results of manual modeling efforts in the domain of systems biology [29]. These efforts of establishing new and reconstructing existing models, used ProBMoT, a software tool for process-based modeling [33]. Through the performed experiments, we have identified a number of design decisions that lead to an improved version of ProBMoT.

In this paper, we perform a systematic evaluation of the latest version of ProBMoT on a cascaded water-tank system [35], a well-known benchmark for nonlinear system identification. In particular, we analyze the robustness of PBM on synthetic data with different noise variances obtained by simulating the benchmark model. Furthermore, we also evaluate the PBM ability to reconstruct a valid model of the system dynamics from measurement data. Therefore, we present here the first proper evaluation of process-based modeling in the context of non-linear system identification.

The remainder of the paper is organized as follows. Section II discusses the relation of process-based modeling to alternative gray-box modeling approaches. Section III introduces the approach of process-based modeling by illustrating it on an example of modeling cascading water tanks.

Section IV elaborates the design of the modeling experiments, the results of which are reported and discussed in Section V. Finally, Section VI concludes the paper with a summary and a brief outline of several directions for further research.

II. RELATED WORK
A. SYMBOLIC REGRESSION APPROACHES
A number of gray-box modeling approaches to system identification aim at both structure and parameter identification of an observed system. Most notably, symbolic regression performed with genetic programming [17] has often been used to approach system identification [15]. The basic idea of performing symbolic regression with genetic programming is to evolve populations of equations, where the fitness function being optimized measures the discrepancy between the observed data and the simulation of the equations. The equations are represented as tree structures, where the internal nodes correspond to mathematical operations and functions, while terminal nodes correspond to system variables and model parameters.

The modeler can control the space of candidate models considered by genetic programming, by specifying the set of primitive operators and functions that can be used in the internal nodes of the evolving trees. In addition, the modeler can also specify a heuristic for model selection that goes beyond the usual degree-of-fit measure mentioned above. Madar *et al.* [21] propose the use of a parsimony principle when designing the fitness function by introducing a penalty for the complexity of the model equations. Rodriguez-Vazquez *et al.* [23] and Ferariu & Patelli [12] propose multi-objective approaches to follow the parsimony principle, where different aspects of degree-of-fit and model complexity are considered as objective functions.

The common limitation of these methods is the coarse control given to the modeler over the space of candidate models. Often, modeling assumptions can not be easily or comprehensibly encoded as inputs to the genetic programming approaches: This leads to a risk of inferring models with implausible structure and/or parameter values. To address this issue, Whigham & Recknagel [34] and Cao *et al.* [7] propose approaches where the space of model structures, considered by genetic programming, is constrained by using grammars [22]. While the grammars allow for flexible specification of the space of candidate models, their use is cumbersome and different for systems' modelers. In addition, the mapping between the structure of the obtained model and that of the observed system is non-trivial to establish: The inference of the mapping is left to the human modeler.

B. COMPONENT-BASED APPROACHES
On the other hand, the mapping between the structure of the model and the observed system is explicit in component-based approaches [2], where models are composed from a library of standardized, well-characterized circuit compo-

nents. Note that the components have fixed constant parameter values, that correspond to physical circuit components and are used for manual construction and design of electrical circuits. No attempts have been made to use libraries of components in the context of computer-aided modeling from knowledge and input/output observational data.

While component-based approaches rely on physical components, block-oriented nonlinear approaches [14] build models out of domain-independent linear, dynamic and nonlinear, static components. In turn, such blocks can be combined into different structures, where each block models a certain aspect of the system dynamics. While some of these combinations are well known and correspond to physical phenomena [27], in principle, different combinations and variations of such components can yield a large number of different structures without a direct mapping to the structure of the observed dynamics at hand. Also, block-oriented approaches usually focus on the manual construction of models, where the block configuration is a priori prescribed by the modeler [26]. Schoukens *et al.* [28] and Schoukens *et al.* [25] propose an approach that requires some modeler's decisions regarding model construction by decomposing the structure identification task. Here, an initial model structure is established first by evaluating the performance of different combinations of linear components using best linear approximation against the nonlinear measured output. The identified initial structure is then used by the modeler as the basis for subsequent steps of adding additional nonlinear blocks to complete the model structure.

C. BLACK-BOX APPROACHES FOR GRAY-BOX MODELING
Recently, some approaches to nonlinear black-box modeling have been also applied in the context of inferring gray-box models. In particular, Brunton *et al.* [6] combine linear regression with the construction of nonlinear terms and Lasso regularization to obtain nonlinear equations-based models that can be considered gray-box. The system SINDy introduces nonlinear terms by using a set of user-specified parameter-free transformations of the original system variables. These nonlinear terms are linearly combined with parameters estimated using simple linear regression. In this respect, SINDy is similar to the early equation discovery system LAGRANGE [11]. A novelty in SINDy with respect to LAGRANGE is the use of Lasso regularization to select terms that are relevant for modeling the observed data. Note, however, that both methods are limited to inferring models with a structure that is linear with respect to the parameters. Genetic programming, grammar-based equation discovery and process-based modeling are more general approaches that can infer models with a structure that is nonlinear with respect to the model parameters.

III. PROCESS-BASED MODELING
Process-based modeling (PBM) is an approach to Equation Discovery that allows the modeler to formalize and use modeling knowledge specific to the domain

of interest. We are going to introduce the PBM approach in the two following subsections. The first subsection introduces a general framework for representing modeling knowledge. In the second subsection, we present the method that employs the formalized knowledge to discover continuous-time models from observation data.

A. FORMALIZATION OF THE MODELING KNOWLEDGE
The PBM framework for formal representation of modeling knowledge is based on the two general concepts of entities and processes. Entities correspond to the static components of the observed system, while processes relate to the dynamic interactions among entities. Each entity includes variables and constants that denote time-varying and time-invariant properties of the corresponding system component, respectively. While entity constants correspond to constant (time-invariant) model parameters, entity variables correspond to the state, input and output variables of the model. The system state is therefore represented with the set of the state variables included in the model entities. On the other hand, each process specifies the interacting entities, process constants and equations. The latter provide the mathematical model of the process influence on the interacting entities, more specifically, on the temporal change, i.e., time derivatives of their state variables. Each process equation can include constants and (both system and input/output) variables of the interacting entities, as well as process constants. When several processes influence the same system variable, the mathematical models of the processes are summed up to obtain the equation that models the dynamics of the system variable at hand.

Figure 1 provides an illustrative example of a process-based model of a dynamic system consisting of two cascaded water tanks. It consists of three entities and three processes as shown in Figure 1b. The three entities correspond to the three system components depicted in Figure 1a: the two tanks and the water pump that fills the upper tank. Each `tank` entity includes a single state (endogenous) variable h, corresponding to the height of the water level in the tank, as well as two constants A and a denoting the areas of the tanks and their effluent areas, respectively. The third entity `pump` includes the input (exogenous) variable v representing the input voltage signal $u(t)$ applied to the pump: the higher the voltage – the higher the water inflow in the upper tank. The voltage-to-flow conversion constant of the pump is denoted with k. The three processes correspond to the entity interactions, which, in this particular example, represent the water flow between the tanks and the pump. The first process INFLOW models the flow from the pump to the upper tank, the second process VALVETRANSFERFUNCTION models the flow from the upper to the lower tank, and the third models the OUTFLOW of water from the lower tank and the system. The latter two processes assume a square-root influence of the height of the water in the tanks on the intensity of the tank valve outflow. The single constant parameter of these two processes specifies the valve transfer constant G that equals $\sqrt{2\,g}$ (where g is the gravitational acceleration).
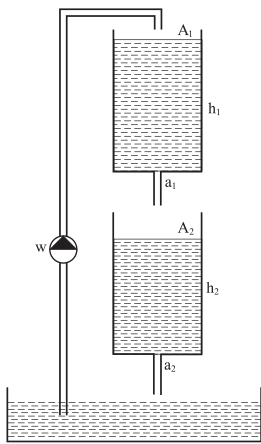
```
entity tank1 : Tank {
    vars: h {role: endogenous; initial: 0.38086;};
    consts:  A = 29.827,
             a = 1.095;}
entity tank2 : Tank {
    vars: h {role: endogenous; initial: 0.20508;};
    consts:  A = 18.93,
             a = 1.123;}
entity pump : Pump {
    vars: v {role: exogenous;};
    consts: k = 7.213;}

process INFLOW(pump,tank1): INFLOW {}
process VALVETRANSFERFUNC (tank1, tank2): SQUAREROOT {
    consts: G=4.429;}
process OUTFLOW(tank2): SQUAREROOT {
    consts: G=4.429;}
```

$$\begin{cases} \dfrac{dh_1}{dt} = -\dfrac{a_1\sqrt{2g}}{A_1}\sqrt{h_1} + \dfrac{k}{A_1}u(t) \\[2ex] \dfrac{dh_2}{dt} = -\dfrac{a_2\sqrt{2g}}{A_2}\sqrt{h_2} + \dfrac{a_1\sqrt{2g}}{A_2}\sqrt{h_1} \end{cases}$$

(a)　　　　　　　　　　　(b)　　　　　　　　　　　(c)

FIGURE 1. (a) The diagram of the three components of the water-tanks system; (b) The process-based model of the water-tanks system; (c) The equation-based model of the water-tanks system.

Process-based models provide additional structure to the model equations with a modular decomposition of the equations into building blocks. These correspond to the knowledge about building models in the domain of interest. In the particular example, the correspondence between the physical structure of the system and the process-based model structure is obvious and explicit. The benefits of this property are twofold. On one hand, it improves the descriptive power of the equations by adding an explanatory layer on top of them. In particular, the equations in Figure 1c do not explicitly reveal the meaning of the repetitive expression $a_1\sqrt{2\,g}\sqrt{h_1}$. In contrast, the process-based model ties that expression to the flow from the upper to the lower tank. On the other hand, and even more importantly, the decomposition into building blocks provides means for formulating the task of inducing a process-based model from data as a combinatorial optimization task. The latter is then constrained to those models that have plausible structure and are therefore acceptable to domain experts. Note, finally, that the process-based models can be easily and automatically rewritten into the standard mathematical form of differential equations that allows for model simulation and analysis.

The process-based formalism for representing modeling knowledge follows the model decomposition into building blocks. In particular, a modeling expert can specify the set of available building blocks in the domain of interest as a library of entity and process templates. The templates represent generic building blocks that, in a particular modeling scenario, can be instantiated into specific components of the model of the observed system. Each entity template prescribes the set of constant and variable properties important for modeling the corresponding static system component

TABLE 1. Library of template entities and processes for modeling an arbitrary configuration of water tanks with three alternative valve transfer functions and three alternative outflow functions.

```
template entity Tank {
    vars: h {aggregation:sum, range:<0,500>};
    consts:  A  {range:<1.0E-3,30>},
             a  {range:<1.0E-3,30>}; }
template entity Pump {
    vars: v;
    consts: k {range:<1.0E-3,30>};}

template process INFLOW(p:Pump, t:Tank){
    equations: td(t.h) = p.k · p.v/t.A; }
template process VALVETRANSFERFUNCTION(t1:Tank, t2:Tank){
    consts : G {range:<0,10>}; //sqrt(2g) }
template process SQUAREROOT:VALVETRANSFERFUNCTION {
    equations :  td(t1.h) = −G · pow(t1.h, 0.5) · t1.a/t1.A,
                 td(t2.h) = G · pow(t1.h, 0.5) · t1.a/t2.A; }
template process LINEAR:VALVETRANSFERFUNCTION {
    equations :  td(t1.h) = −G · t1.h · t1.a/t1.A,
                 td(t2.h) = G · t1.h · t1.a/t2.A; }
template process EXPONENTIAL:VALVETRANSFERFUNCTION {
    equations :  td(t1.h) = −G · exp(t1.h) · t1.a/t1.A,
                 td(t2.h) = G · exp(t1.h) · t1.a/t2.A; }
template process OUTFLOW(t:Tank) {
    consts : G {range:<0,10>};
template process SQUAREROOT:OUTFLOW {
    equations: td(t.h) = −G · pow(t.h, 0.5) · t.a/t.A;}
template process LINEAR:OUTFLOW {
    equations: td(t.h) = −G · t.h · t.a/t.A;}
template process EXPONENTIAL:OUTFLOW {
    equations: td(t.h) = −G · exp(t.h) · t.a/t.A;}
```

in the domain. Similarly, the process templates are generic interactions and prescribe template mathematical expressions for modeling them.

Table 1 provides an example library of template entities and processes for modeling systems of connected water tanks

and pumps. The two generic entities of `Tank` and `Pump` can represent an arbitrary component of any system in the water-tanks domain. The entities `tank1` and `tank2` in the example process-based model from Figure 1b represent two specific instances of the template entity `Tank`. Note that the modeling knowledge encoded in the template entities include specifications of the constant and variable properties of the corresponding system components with their operational ranges rather than specific values. The `aggregation: sum` declaration specifies that when an entity `Tank` participates in more than one process, the multiple influences of the different processes on the entity variable `h` are summed up.

The template processes are organized in a hierarchy. In Table 1, the VALVETRANSFERFUNCTION template represents the root node of a hierarchy with three descendants, i.e. SQUAREROOT, LINEAR and EXPONENTIAL, that correspond to three alternative mathematical models of valve transfer. The first assumes the square-root, the second linear and the third exponential influence of the water level in the tank on the tank-valve outflow. Similarly, the OUTFLOW hierarchy of template processes specifies three alternative models of water flow from the system to its environment.

The process-based library provides a general recipe for building models in the domain of use. Following the encoded knowledge, models of arbitrary complexity can be established: They can include an arbitrary number of entities and interactions among them. In particular, the library from Table 1 can be re-used for modeling water-tanks systems with an arbitrary number and configurations of tanks, pumps and flows among them and the system environment. Such knowledge also allows the modeler to explore various modeling assumptions. In situations where the valve transfer function is known, one can prescribe that the specific process should be an instance of, e.g., the SQUAREROOT template. On the other hand, if the valve transfer function is unknown and should be induced from the observational data, one should prescribe the root template of the hierarchy, i.e., VALVETRANSFERFUNCTION.

The alternative configurations of model components and corresponding mathematical expressions can be automatically explored by the algorithm for inducing process-based models from data and knowledge, introduced in the following subsection.

B. INDUCING PROCESS-BASED MODELS

Figure 2 outlines the ProBMoT algorithm for inducing process-based models, which takes three inputs. The first input is the process-based library introduced in the previous subsection. The second input specifies the modeling scenario including the specific entities that appear in the observed system, the partially known configuration of the interactions among them and the modeling assumptions. The third is the training and validation data sets of measurements of the state and input/output variables of the observed system. Given its inputs, ProBMoT proceeds in three steps: (1) instantiating templates from the library into specific model components,

(2) generating candidate model structures corresponding to combinations of the model components, and (3) estimating constant parameters and the fit of each model structure against the data. Finally, the output of ProBMoT is a list of continuous-time process-based models ranked according to their performance, e.g. the discrepancy between the model simulation and the validation data set.

In the first step, given the entities of the observed system and the modeling assumptions, each template process is instantiated into a number of specific instance processes. For example, given two tank entities, the template process VALVETRANSFERFUNCTION has six instances. The first three of them correspond to the situation where the water flows from the first to the second tank, with each of the three corresponding to one valve transfer form. The remaining three correspond to the flows from the second tank to the first. Table 2 introduces a specific modeling scenario for an observed system consisting of a cascade of a pump and two water tanks: the pump fills the first tanks, there is a flow between the first and the second tank, and the water of the second tank outflows into the system environment. In this particular scenario, the INFLOW generic process has a single instance and each of the VALVETRANSFERFUNCTION and OUTFLOW processes has three instances.

The seven individual process instances represent the components for building process-based models of the observed system in the second step of the algorithm. Finding an optimal model is equivalent to the task of finding an optimal combination of the model components. This is a task of combinatorial optimization, which can be approached by an arbitrary algorithm for combinatorial search. In the particular modeling scenario, from the seven model components, we need to compose a model that has an INFLOW, VALVETRANSFERFUNCTION and an OUFLOW component. This leads to $1 \times 3 \times 3 = 9$ candidate combinations, so we employ exhaustive search, i.e., enumerate all the candidates. Alternatively, when facing more complex modeling scenarios leading to large search spaces, one can employ a scalable method for combinatorial optimization that performs incomplete, heuristic search through the space of candidate combinations. To this end, symbolic regression methods often employ evolutionary approaches, such as genetic algorithms [24], while equation discovery methods typically employ greedy search algorithms [31].

In the final (third) step of the algorithm, ProBMoT evaluates each candidate model structure by estimating the values of its constant parameters. Recall that the list of constant parameters and the ranges of their values are included in the process-based library. ProBMoT approaches parameter estimation as a numerical optimization task. The objective function for numerical optimization is the output error, i.e., the discrepancy between the simulated model response and the measured system response in the training data set [4], defined as:

$$RRMSE(y) = \sqrt{\frac{\sum_{t=0}^{n}(y_t - \hat{y}_t)^2}{\sum_{t=0}^{n}(\bar{y} - \hat{y}_t)^2}}. \tag{1}$$
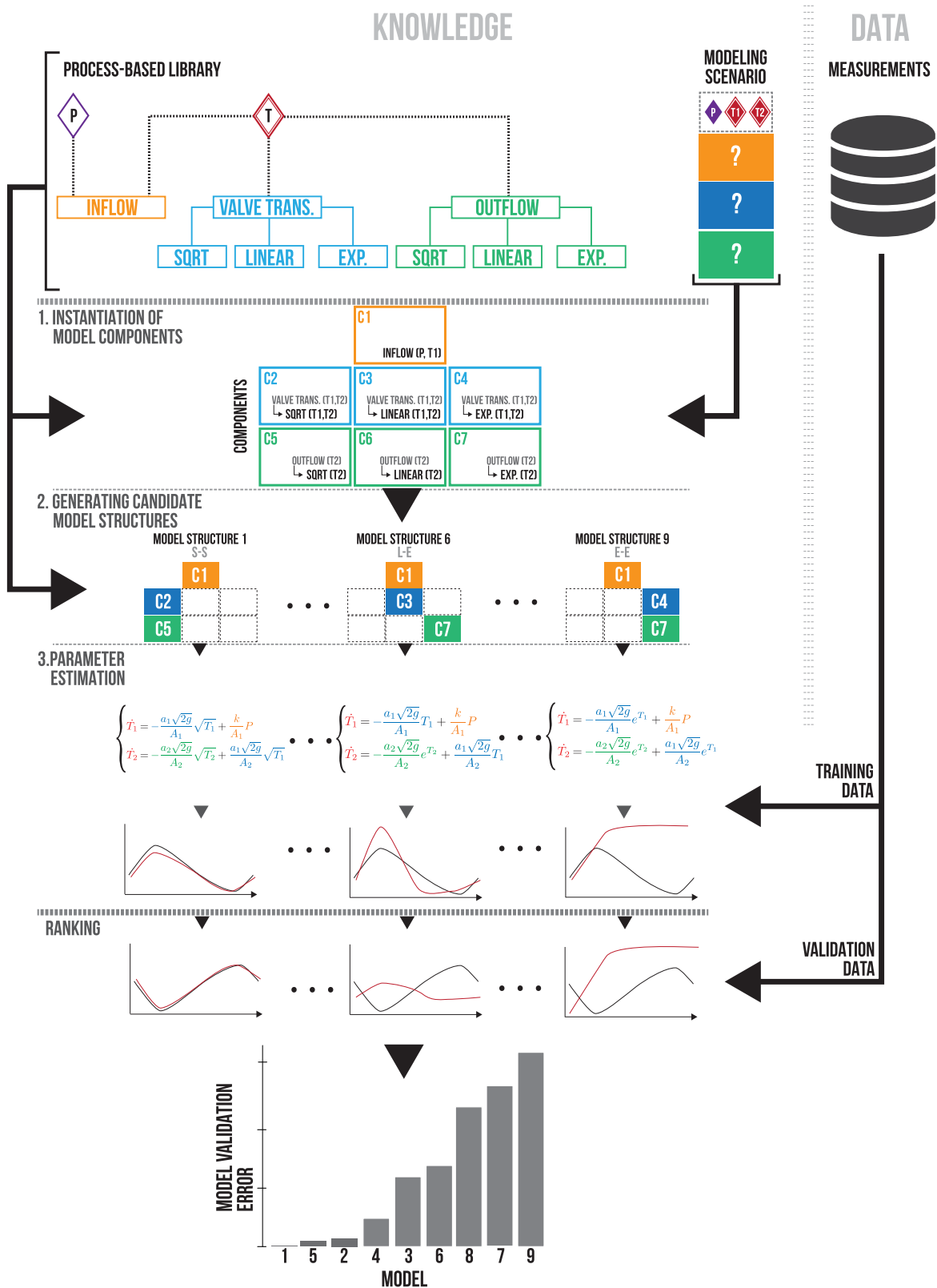
FIGURE 2. Workflow of the ProBMoT algorithm for inducing process-based models from knowledge and data. The knowledge specifies two types of entities (P and T) that refer to a pump and a tank, as well as three types of processes (INFLOW, VALVETRANSFERFUNCTION and OUTFLOW). Given the process-based library and the modeling scenario, 9 candidate models structures are constructed, listed in Table 3. These structures are mapped to equations, have their parameters estimated and are finally ranked based on model validation error.

TABLE 2. Modeling scenario for the system of two cascaded water tanks and a pump.

```
entity tank1 : Tank {
    vars: h {role: endogenous; initial: 0.38086;};
    consts:  A = null,
             a = null;}
entity tank2 : Tank {
    vars: h {role: endogenous; initial: 0.20508;};
    consts:  A = null,
             a = null;}
entity pump : Pump {
    vars: v {role: exogenous;};
    consts: k = null;}


process INFLOW(pump,tank1): INFLOW {}
process VALVETRANSFERFUNC(tank1, tank2):
    VALVETRANSFERFUNCTION { consts: G=4.429;}
process OUTFLOW(tank2): OUTFLOW { consts: G=4.429;}
```

In Eq. (1) n denotes the number of measurements, y_t and $\hat{y}_t$ correspond to the *measured* and *simulated* values of the output variables y and $\bar{y}$ denotes the mean value of y in the data set. Note that the *RRMSE* takes the total squared error (the numerator of the fraction on the right-hand side of Eq. (1)) and normalizes it by dividing it with the total squared error of the trivial model that predicts the average of the observed values. Thus, the model with the *RRMSE* of 1 has the same error as the error of the trivial model, while smaller values indicate the relative improvement over it.

ProBMoT employs the CVode library [8] for simulating continuous-time models in the form of ordinary differential equations. CVode implements a general-purpose ODE solver with linear multi-step variable-coefficient methods for integration. Note that we perform long-term simulation of the model equations given the initial conditions only: This is in contrast to most simulation approaches that rely on value-corrections of the simulation response at each simulation step [36]. Given that the objective function includes simulation of non-linear continuous-time models and an output-error objective function, ProBMoT employs nonlinear methods for numerical optimization. In particular, it uses the Differential Evolution algorithm as implemented in the metaheuristic optimization framework jMetal [10]. Differential evolution is arguably a powerful stochastic optimization algorithm with encouraging results related to its asymptotic convergence towards global optima [13].

IV. EXPERIMENTAL DESIGN
To properly test the hypothesis that the process-based modeling approach can accurately perform nonlinear system identification, we propose investigating the two aspects of (1) structure identification and (2) parameter estimation. To this end, the experiments consider modeling a cascading water tank system with two tanks as shown in Figure 1a, where we focus on modeling the dynamics of the lower tank h_2. In the first line of experiments, we investigate

ProBMoT's ability to discriminate among different instantiated candidate structures. Next, we analyze its ability to correctly reconstruct the values of the model parameters.

In particular, we consider two modes of operation of ProBMoT: (1) multi-stage mode – MULTI-STAGE PROBMOT and (2) single-stage mode – SINGLE-STAGE PROBMOT. The former relates to a standard approach in system identification where the problem is decomposed so that the dynamics of each signal is modeled separately [25]. Namely, using ProBMoT we first attempt to establish the structure and parameters of the model of the first tank, using the signal h_1 in the learning process. In turn, the best performing (intermediate) model is used for establishing the structure and parameters of the lower-tank model using the signal h_2. On the other hand, SINGLE-STAGE PROBMOT refers to the standard operation of ProBMoT, where it attempts to learn the complete model for both signals h_1 and h_2 simultaneously.

Regarding the data, in our experiments, we consider both synthetic data, by varying the values of the signals h_1 and h_2 at each time point, as well as measured data. The latter consists of 2500 measurements of the input signal $u(t)$ and the two output signals, i.e., the water levels of the two tanks, h_1 and h_2. Further details on the measurement procedures performed on a real water-tank system are given by Wigren and J. Schoukens [35]. To properly evaluate model performance, we split these data into a train, a validation, and a test set, consisting of 1000, 500 and 1000 measurements, respectively. In particular, we learn the model structure and parameters using the train data, we select an optimal model with respect to its performance on a validation data set and evaluate the performance of the selected model on the test set.

The synthetic data is generated by simulating the model from Fig 1b using arbitrarily selected parameter values of $a_1 = 0.65$, $A_1 = 20$, $a_2 = 0.7$, $A_2 = 12$ and $k = 5$. For the input signal $u(t)$, we used the real measurements, while for h_1 and h_2, we used the simulated responses with Gaussian noise added following the equation $y_{noise}(t) = y(t) \cdot (1 + N(0, v))$, where $N(0, v)$ denotes a normal-distribution random variable with mean 0 and variance v. Note that we do not add noise to the input signal, since it comes from the measurements. We generate five different train/validation sets of synthetic responses with variance of 0.01, 0.02, 0.05, 0.1 and 0.2, where the uncorrupted responses correspond to variance 0. This yields six different synthetic experimental cases/data sets.

In both the single-stage and the multi-stage modes, ProBMoT uses the process-based library of domain knowledge for modeling cascading-tank dynamics, presented in Table 1. The library combined with the modeling scenario given in Table 2, yields 9 candidate model structures shown in Table 3. For instance, S-S denotes the model where the processes VALVETRANSFERFUNCTION and OUTFLOW have sublinear SQUAREROOT dynamics. Note that, the artificial data was generated with the model S-S with the parameter values stated in the previous paragraph. The different modeling scenarios used in the experiments with single-stage and multi-stage ProBMoT are given in Appendix.

TABLE 3. The list of nine candidate model structures considered by ProBMoT. The acronyms correspond to the different components included in the respective interaction. For example, Model S-S denotes a model structure with SQUAREROOT modeling alternative for both processes, VALVETRANSFERFUNCTION and OUTFLOW. All model structures include the INFLOW process.

VALVETRANSFERFUNC OUTFLOW	SQUAREROOT SQUAREROOT	SQUAREROOT LINEAR	SQUAREROOT EXPONENTIAL	LINEAR SQUAREROOT	LINEAR LINEAR	LINEAR EXPONENTIAL	EXPONENTIAL SQUAREROOT	EXPONENTIAL LINEAR	EXPONENTIAL EXPONENTIAL
Model	S-S	S-L	S-E	L-S	L-L	L-E	E-S	E-L	E-E

Furthermore, the following parameter settings for Differential Evolution were used: a population size of 60, strategy rand/1/bin, differential weight (F) and a crossover probability (Cr) of 0.9. The limit on the number of evaluations of the objective function is $5 \cdot 10^4$ per parameter. This particular parameter setting of DE is based on previous studies [30]. Finally, for simulating the models, we used the CVode simulator [8] with a standard setting of absolute and relative tolerances, 10^{-8} and 10^{-4}, respectively. For generating the synthetic data as well as perform model simulation in ProBMoT, we use the backward differentiation method combined with Newton iteration and a preconditioned Krylov method with 10^5 steps as implemented in CVODE [8].

V. RESULTS

We investigate the ability of ProBMoT to reconstruct the correct models in both a synthetic and a real-world setting. In the first line of experiments, we examine ProBMoT's ability to recover the ground-truth model S-S from synthetic data. In particular, we present performance comparisons between the two modes of ProBMoT, MULTI-STAGE ProBMoT and SINGLE-STAGE ProBMoT, in terms of $RRMSE_{validation}$ and $RRMSE_{test}$ on the six synthetic experimental cases. Next, we investigate the ability of the two modes of ProBMoT to estimate the ground-truth parameters from the synthetic data. In the second line of experiments, we test the ability of ProBMoT to identify the structure and parameter of the model from measurements of the water-tanks system.

A. RESULTS ON SYNTHETIC DATA

Figure 3 depicts the errors of the models learned with MULTI-STAGE ProBMoT. The top two graphs in Figure 3 refer to the errors of the three "intermediate" candidate models of the dynamics of h_1 considered in the first stage. Figure 3a shows that ProBMoT can identify the structure of the process VALVETRANSFERFUNCTION for h_1 correctly. Moreover, Figure 3b shows that the identified intermediate model with a SQUAREROOT alternative of the process VALVETRANSFERFUNCTION for the top tank has a constant and substantially better test performance than the other two models (with the LINEAR or EXPONENTIAL alternative).

The two bottom graphs in Figure 3 depict the errors of the three final models that complete the correctly identified intermediate model of h_1 in the first stage. Figure 3c shows that ProBMoT selects the correct complete model S-S for all noise variances. In particular, the validation error of the model S-S in the noise-free scenario is almost 40 percentage
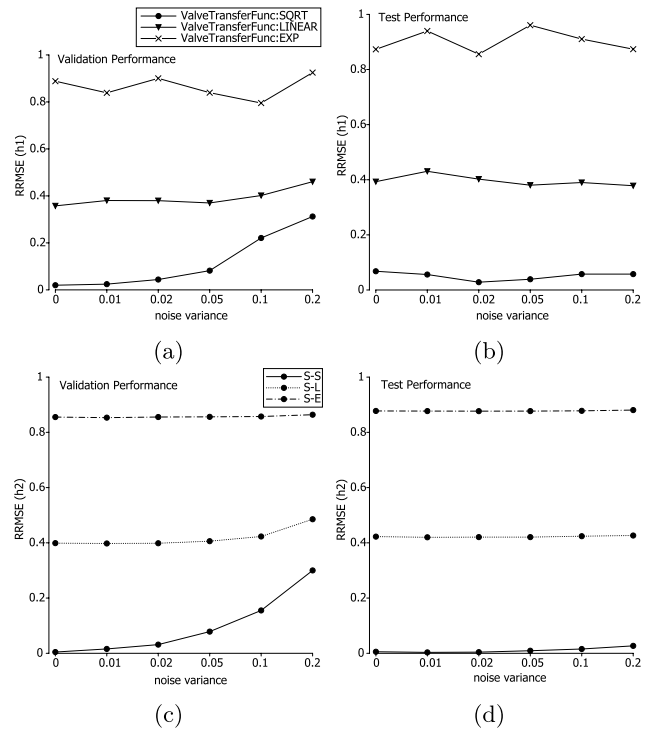

FIGURE 3. MULTI-STAGE ProBMoT: **Top-** Performance comparison of the simulation error (*RRMSE*) of the h_1 signal obtained from the 3 intermediate model structures on the (a) validation and (b) test synthetic data sets in the six different synthetic cases. **Bottom-** Performance comparison of the simulation error (*RRMSE*) of the output signal h_2 obtained from the remaining 3 model structures on the (c) validation and (d) test synthetic data sets in the six different synthetic cases.

points lower than the second-ranked model S-L, and almost 20 percentage points lower for the 0.2 noise variance. This result is also in-line with the test performance of the S-S model, depicted in Figure 3d, which, in general, performs up to 30 percentage points better than the second-ranked model S-L and up to 90 percentage points better than the third-ranked S-E.

While MULTI-STAGE ProBMoT is able to correctly identify the ground-truth model, it follows a procedure that requires an intermediate intervention by the modeler. The decomposition and the ordering of stages are to be provided by a human expert that, in turn, has to run ProBMoT for each stage separately. In contrast, SINGLE-STAGE ProBMoT does not require any intervention by a modeler. Its identification capabilities are presented in Figure 4. SINGLE-STAGE ProBMoT explores the complete space of nine candidate model structures, using
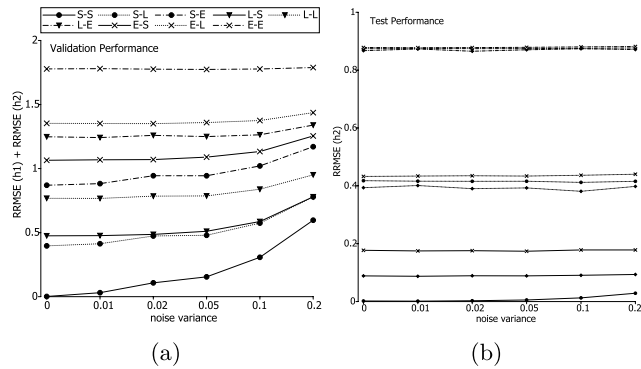
FIGURE 4. Single-stage ProBMoT: (a) Performance comparison of the summed validation errors of both signals h_1 and h_2 ($RRMSE_{h1} + RRMSE_{h2}$) from the 9 obtained models across 6 different synthetic cases. (b) Performance comparison test errors of the output signal h_2 ($RRMSE_{h2}$) from the 9 obtained models across 6 different synthetic cases.
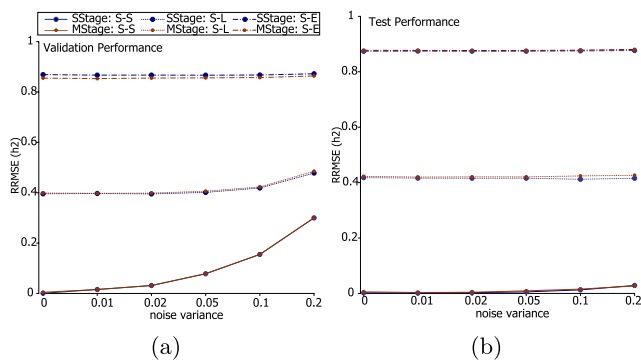

FIGURE 5. comparison of the errors of the three candidate models considered with Single-stage ProBMoT (SStage) and Multi-stage ProBMoT (MStage), measured on the response of the output signal h_2.

both signals h_1 and h_2 as input, and ranks the models according to the summed error $RRMSE_{h1} + RRMSE_{h2}$.

Figure 4a shows that Single-stage ProBMoT is able to reconstruct the correct model structure S-S: In particular, based on the performance on a separate validation set, Single-stage ProBMoT accurately identifies the correct structure for all 6 synthetic cases. Note that here the relative difference in performance between the best-identified structure and the remaining structures is substantial and higher than 20 percentage points. Moreover, in terms of performance on unseen test data, Single-stage ProBMoT exhibits stable behavior, where for all synthetic cases the discrepancy between the model's response of the output signal h_2 and the actual response (in the synthetic test data) remains stable and lower than 0.05.

In sum, both modes of using ProBMoT accurately identify the correct structure of the model. Figure 5 presents a comparison of the errors of the three candidate models considered with Multi-stage ProBMoT and Single-stage ProBMoT, measured on the output signal h_2. Figure 5 shows that both methods lead to models with virtually identical errors on both validation (Figure 5a) and test (Figure 5b) data. This means

that for the task of structure identification, Single-stage ProBMoT without any intervention by a modeler, explores the complete (and larger) space of candidate model structures, and performs almost identically to Multi-stage ProBMoT. The latter embodies a typical approach in system identification, where the modeling task is tackled via decomposition into multiple stages and requires intervention by a modeler.

Finally, we assess the capability of both Multi-stage ProBMoT and Single-stage ProBMoT to reconstruct the values of the model parameters from noisy data. Figure 6 depicts the relative difference between the estimated parameter values in the learned models and the original parameter values used for generating the synthetic data. To obtain reliable parameter estimates, we performed 100 repetitions of parameter estimation with Differential Evolution (by varying the random seed) for each of the six synthetic cases. Note that, while the model structure has 5 parameters, in Fig 6, we present the relative errors for the constants that appear in the equation from Fig 1c, i.e. the ratios of the parameters a_1/A_1, k/A_1, a_2/A_2 and a_1/A_2, respectively.

As we can see from Fig 6, both modes of using ProB-MoT can accurately estimate the original parameter values regardless of the noise variance. In general, the relative difference between the estimated and the original parameter values for both methods is lower than 5% across all experimental cases. Note, however, that Single-stage ProBMoT performs much better, lowering the variance of the parameter estimates to below 1%. In the learning phase, Single-stage ProBMoT considers models with more degrees of freedom allowing more parameters in the model structures to be estimated simultaneously leading to a better fit. In the case of Multi-stage ProBMoT, while the final model has the same number of parameters, it is sequentially constructed from models with fewer parameters. Such an approach constraints the parameter search space (notably in the latter learning stages) which ultimately affects the fit. Moreover, any (likely) fitting errors that accumulate in the former stages are also propagated to the subsequent stages, thus impairing the final performance. Nevertheless, for the particular modeling task at hand, the differences in the parameter values obtained with the two modes of ProBMoT are not substantial, since both sets of parameter values can accurately model the dynamics of the water-tanks system.

B. RESULTS ON MEASURED DATA

Here we focus on testing the ability of the process-based modeling approach to accurately model the benchmark system from measured data. Note that, for this modeling attempt, we assume that the correct structure of the system corresponds to the one given in Figure 1c while the parameters are unknown and remain to be found. More specifically, since both Multi-stage ProBMoT and Single-stage ProBMoT performed identically on the synthetic experiments, here we only use Single-stage ProBMoT to find the structure of the model and estimate the model's parameters. Table 4 presents the results of this experiment.
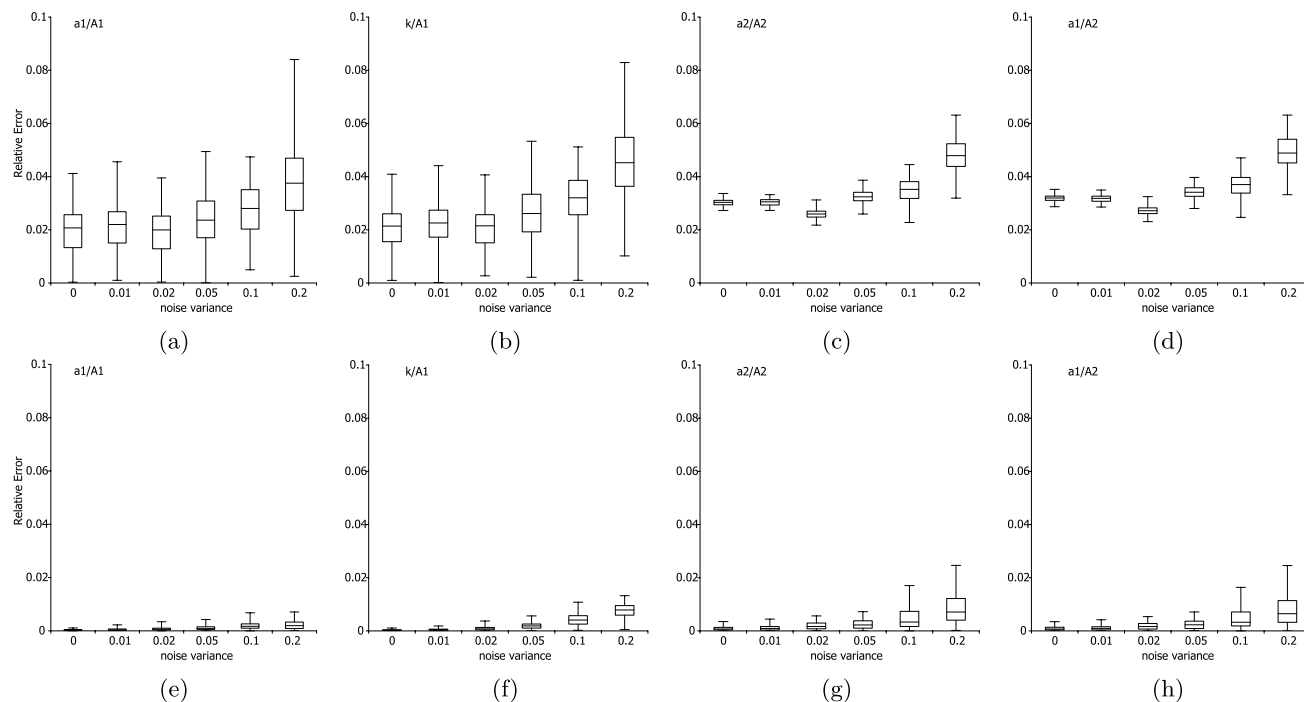
FIGURE 6. Relative error between the ground-truth and the fitted parameter values estimated at various levels of noise using (**Top**) MULTI-STAGE PROBMOT and (**Bottom**) SINGLE-STAGE PROBMOT. Here the relative errors are measured on the ratios of the 5 parameters : (a,e) a_1/A_1, (b,f) k/A_1, (c,g) a_2/A_2 and (d,h) a_1/A_2.

TABLE 4. The summed validation error for both signals h_1 and h_2 ($RRMSE_{h1} + RRMSE_{h2}$) and the test error for the output signal h_2 ($RRMSE_{h2}$) from the 9 models obtained from measured data.

Model	$RRMSE_{h1} + RRMSE_{h2}$ VALIDATION	$RRMSE_{h2}$ TEST
S-S	*0.481*	**0.23**
S-L	0.539	0.306
S-E	1.038	0.805
L-S	**0.44**	*0.249*
L-L	0.494	0.313
L-E	0.999	0.805
E-S	0.977	0.248
E-L	1.125	0.37
E-E	1.592	0.818

From Table 4, we can see that SINGLE-STAGE PROBMOT struggles to identify the assumed correct structure from the measured data. Namely, we can see that the L-S model performs better than the S-S model in terms of the summed *RRMSE* for both signals. Nevertheless, in terms of test performance, the S-S model has the best performance on the test set with the model L-S being second. Note, however, that the difference in test performance between S-S and L-S remains low and insignificant, even if in favor of the model S-S. To properly assess the behaviors of the two models, Fig 7 presents the simulation responses on the test data of the S-S and L-S models. From the figure, we can see that the simulated responses are indistinguishable. We conjecture that this is due to implicit noise in the data, possibly due to
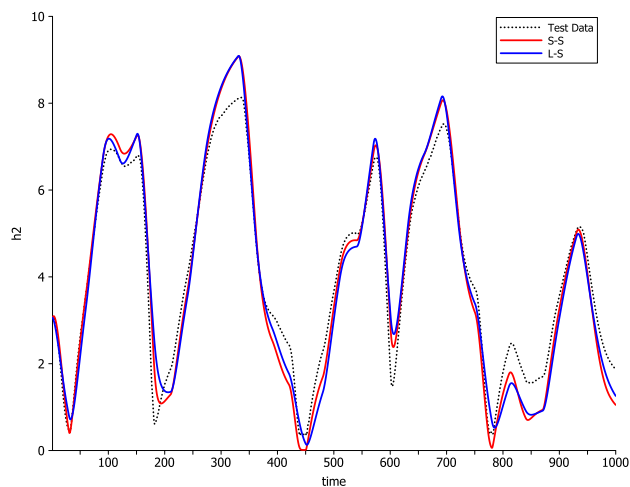

FIGURE 7. Simulation responses on the test data of the two top-ranked models S-S and L-S obtained from measured data.

unforeseen nonlinearities in the measuring set-up, which have not been included in the knowledge provided at input.

C. DISCUSSION

In general, the results from the empirical analyses show that process-based modeling can be successfully applied to the task of nonlinear systems identification. In particular, from the experiments performed in the synthetic cases, we can conclude that ProBMoT, using domain knowledge and data,

can accurately identify the correct structure of the system as well as estimate the model's parameter values with high accuracy, regardless of the signal corruption present in the data. In the measured-data experiments, however, ProBMoT performs moderately well by discarding the majority of the models with incorrect structures, but nevertheless struggling to identify the ''correct'' one among the top two ranked. We conjecture that such a performance might be an artifact of nonlinearities in the measuring set-up, not captured in the domain knowledge provided at input. Such a lack of knowledge can be explicitly handled by the modeler, where one can additionally decide on the correct structure. However, in the case of process-based modeling, such behavior can also be handled by extending the modeling knowledge provided at input: This can be achieved either by encoding new processes in the library of domain knowledge or by weakening some of the constraints that relate to the dynamics of the individual processes.

For this particular task, we extend the modeling knowledge by weakening the constraint on the two processes VALVETRANSFERFUNCTION and OUTFLOW. Given that ProB-MoT mostly struggled to differentiate between strictly linear and nonlinear (square-root) components for the interactions that involve the two tanks h_1 and h_2, we add a component that allows for all sub-linear interactions. We encode this in the process-based library Table 5, by adding a new process alternative POWER to each of the processes VALVETRANSFERFUNCTION and OUTFLOW. These two alternatives in turn are instantiated to two additional components which allow for modeling any sub-linear dynamics in VALVETRANSFERFUNCTION and OUFLOW. In particular, both processes introduce two additional parameters P_{h1} and P_{h2}, where:

$$0 < P_{h1}, P_{h2} < 1,$$

which correspond to the exponents of the two signals h_1 and h_2, respectively, in the equations. Table 6 presents the results of the modeling experiment given the extended process-based library. To test this extension, we perform experiments using measured data, where we evaluate the performance of a new model P-P and compare its performance against the models S-S and L-S.

The results reported in Table 6 show that the model P-P performs substantially better on the validation set ($RRMSE_{h1} + RRMSE_{h2}$) than both the S-S and L-S models. This result confirms our hypothesis that, for this particular modeling task, extending the modeling knowledge allows for better identification. Note that the estimated parameter values of P_{h1} and P_{h2} for the signals h_1 and h_2 are 0.68 and 0.66, respectively. This result is also in-line with our hypothesis that there are some implicit sub-linear dynamics captured in the measured data, which does not correspond to the original theoretical model from Figure 1c. Moreover, Figure 8 presents the simulated responses on the test set of the three models P-P, S-S and L-S. From the figure, we can see that, for some of the trends where S-S and L-S struggle, P-P is

TABLE 5. Extension of the library for cascaded water tank systems with new process alternatives.

⋮

```
template process POWER:VALVETRANSFERFUNCTION {
    equations :
        td(t1.h) = −G · pow(t1.h, P) · t1.a/t1.A,
        td(t2.h) = G · pow(t1.h, P) · t1.a/t2.A; }
```

⋮

```
template process POWER:OUTFLOW {
    equations:
        td(t.h) = −G · pow(t.h, P) · t.a/t.A; }
```

⋮

TABLE 6. The summed validation errors of both signals h_1 and h_2 ($RRMSE_{h1} + RRMSE_{h2}$) and test errors of the output signal h_2 ($RRMSE_{h2}$) from the new established model P-P and the two models S-S and L-S constructed from measured data.

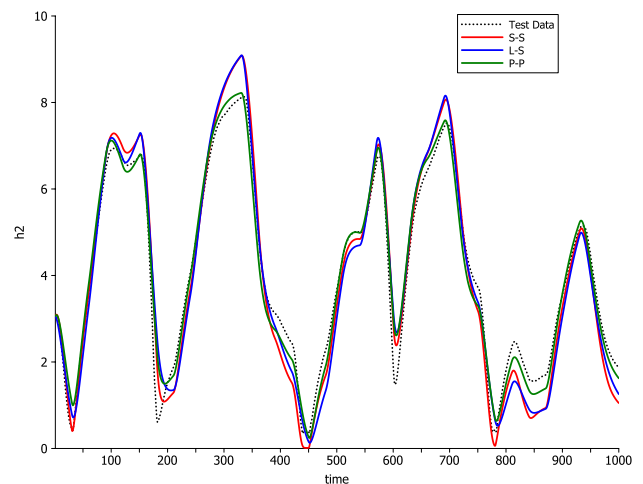| Model	$RRMSE_{h1} + RRMSE_{h2}$ VALIDATION	$RRMSE_{h2}$ TEST
S-S	0.481	0.23
L-S	0.44	0.249
P-P	**0.238**	**0.164**

FIGURE 8. Simulation responses on the test data of the constructed models P-P, S-S and L-S obtained from measured data.

able to capture them with high accuracy. In turn, this accuracy translates to lower predictive error on the output signal h_2 of $RRMSE_{h2} = 0.164$ in the case of P-P, as compared to the errors $RRMSE_{h2} = 0.23$ of the model S-S and $RRMSE_{h2} = 0.249$ of the model L-S.

In sum, we have shown that by weakening the constraints of the processes VALVETRANSFERFUNCTION and OUTFLOW in the process-based library, ProBMoT is able to discriminate among the model structures P-P, on the one hand, and the remaining two model structures, on the other. This yields more accurate models of the benchmark system from measured data. Note that, while we implemented this extension by
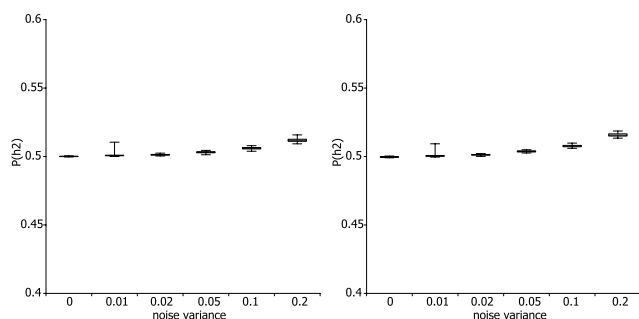
FIGURE 9. Estimated values of the two parameters P_{h1} and P_{h2} across the six synthetic data sets.

adding two additional parameters, the performance improvement is a direct result of changing the structure of the model. To test this hypothesis, we performed an additional parameter estimation experiment for estimating the values of P_{h1} and P_{h2} using the synthetic data sets. For this task, we performed 100 repetitions for each of the 6 different synthetic cases. The results for both parameters are given in Figure 9.

From Figure 9, we can see that ProBMoT estimates the values of both parameters to be in the range from 0.5 (with standard deviation $\sigma = 10^{-4}$), in the synthetic case with noise variance 0, to 0.51 (with standard deviation $\sigma = 10^{-3}$) in the case with noise variance 0.2. This stable behavior means that, in a controlled scenario where the appropriate amount of knowledge is present in the process-based library, ProBMoT is able to identify a structure identical to the underlying correct model structure (S-S) regardless of the corrupted responses of the input/output signals.

VI. CONCLUSION

In this paper, we present the first proper evaluation of process-based modeling in the context of system identification. Process-based modeling, a recent equation discovery approach, addresses the task of computer-aided modeling of nonlinear dynamic systems from measurements and domain-specific modeling knowledge. We perform a systematic evaluation of its utility for reconstructing a standard system identification benchmark model of a cascaded water-tank system. To this end, we test the latest implementation of process-based modeling, i.e. the ProBMoT system, which allows for computer-aided construction, long-term simulation and evaluation of process-based models of dynamic systems. More specifically, we first analyze the robustness of ProBMoT on a series of identification tasks involving synthetic data with different noise variances, obtained by simulating the benchmark model. Next, we investigate ProBMoT's ability to establish a valid model of a real benchmark system from measurement data.

The empirical evaluation shows that ProBMoT can reconstruct both the structure and the parameters of a system from synthetic data with various levels of corruption, as well as from measured data. In particular, in the synthetic-data experiments, ProBMoT is able to accurately identify the correct structure of the system, as well as to estimate the

model's parameter values with high accuracy, regardless of the signal corruption present in the data. On the other hand, in the measured-data experiments, where ProBMoT initially struggled to discriminate between the two top-ranked models, we were able to identify implicit nonlinearities that were present in the data and not in the modeling knowledge provided at input. Consequently, with a minimal extension of the process-based library, we showed that ProBMoT can successfully reconstruct both the structure and the parameters of the model from measured data.

The process-based modeling approach, and in particular its application to the task of nonlinear system identification, is the main contribution of this paper. Namely, PBM has several distinguishing properties. First, PBM is a "white-off" gray-box modeling approach [20] that constructs models from components with a clear mapping to the modeled physical phenomena. Next, the resulting process-based models retain the utility of mathematical models and therefore can be simulated and analyzed using well established numerical approaches. Finally, PBM is general and can be, in principle, applied to any domain and to any other task of system identification. Moreover, the process-based modeling knowledge employed by PBM is domain-specific in the sense that it captures the basic modeling principles in a given domain and can be reused for different modeling applications within the same domain.

Note, however, that the necessity to formalize process-based modeling knowledge for the domain of interest can also represent a serious obstacle for application of the PBM approach. Domain-specific knowledge about decomposition of the models of dynamical systems into entities and processes might be unavailable or out of reach of the user of ProBMoT. In this context, general symbolic regression approaches [6], [24] are much easier to apply to a new domain, since they are purely data-driven and do not require formalization of domain-specific knowledge. Note, however, that, in many cases, the equations obtained with symbolic regression approaches are unnecessarily complex and often hard to comprehend by the domain experts, since they do not make direct contact with the existing knowledge and models. Moreover, PBM employs nonlinear optimization methods for parameter estimation, which might lead to suboptimal values of the constant parameters of the model. An alternative approach would be to adapt (sparse) linear regression methods suggested in [6]. This would, however, impose a requirement that the process-based model structures are linear with respect to the parameters, which might obscure the clear mapping between the model processes and the physical processes of the modeled phenomena.

Several directions for further work can be followed. While our empirical study is limited to one task, i.e. modeling the dynamics of a cascaded water tank system with two tanks, the proposed approach to learning process-based models can be applied to many other tasks of system identification, both from the same domain as well as from other domains. An immediate continuation of the work presented here would

be to investigate whether the findings of this paper also apply to other standard nonlinear benchmark systems in both real-world and synthetic settings. Next, we intend to compare the process-based modeling approach to other state-of-the-art gray-box modeling approaches, both in terms of accuracy of reconstructing dynamics of a nonlinear system and in terms of the comprehensibility of the models that these approaches can provide.

TABLE 7. Modeling scenario with SINGLE-STAGE PROBMOT. The initial values of h_1 and h_2 are taken from the measured data.

```
entity tank1 : Tank {
    vars: h {role: endogenous; initial: 0.38086;};
    consts :  A = null,
              a = null; }
entity tank2 : Tank {
    vars: h {role: endogenous; initial: 0.20508;};
    consts :  A = null,
              a = null; }
entity pump : Pump {
    vars: v {role: exogenous;};
    consts: k = null;}


process INFLOW(pump,tank1): INFLOW {}
process VALVETRANSFERFUNC(tank1, tank2):
    VALVETRANSFERFUNCTION { consts: G=4.429;}
process OUTFLOW(tank2): OUTFLOW { consts: G=4.429;}
```

TABLE 8. Modeling scenario with MULTI-STAGE PROBMOT, for both the first stage (top) and the second stage (bottom). The initial values of h_1 and h_2 are taken from the measured data, while the parameter values of the upper tank in Stage 2 are the outcome of Stage 1.

```
//STAGE 1
entity tank1 : Tank {
    vars: h {role: endogenous; initial: 0.38086;};
    consts :  A = null,
              a = null; }
entity pump : Pump {
    vars: v {role: exogenous;};
    consts: k = null;}


process INFLOW(pump,tank1): INFLOW {}
process VALVETRANSFERFUNC(tank1):
    VALVETRANSFERFUNCTION {consts: G=4.429;}
//STAGE 2
entity tank1 : Tank {
    vars: h {role: exogenous; initial: 0.38086;};
    consts :  A = 23.98,
              a = 0.762; }
entity tank2 : Tank {
    vars: h {role: endogenous; initial: 0.20508;};
    consts :  A = null,
              a = null; }
process VALVETRANSFERFUNC(tank1, tank2):
    VALVETRANSFERFUNCTION { consts: G=4.429;}
process OUTFLOW(tank2): OUTFLOW {consts: G=4.429;}
```

APPENDIX
MODELING SCENARIOS
See Tables 7 and 8.

REFERENCES

[1] D. Aleksovski, J. Kocijan, and S. Dzeroski, "Ensembles of fuzzy linear model trees for the identification of multioutput systems," *IEEE Trans. Fuzzy Syst.*, vol. 24, no. 4, pp. 916–929, Aug. 2016.

[2] P. J. Ashenden, *The Designer's Guide to VHDL*, 3rd ed. San Mateo, CA, USA: Morgan Kaufmann, 2008.

[3] N. Atanasova, L. Todorovski, S. Džeroski, Š. R. Remec, F. Recknagel, and B. Kompare, "Automated modelling of a food Web in lake Bled using measured data and a library of domain knowledge," *Ecolog. Model.*, vol. 194, nos. 1–3, pp. 37–48, Mar. 2006.

[4] L. Breiman, J. H. Friedman, C. J. Stone, and R. A. Olshen, *Classification Regression Trees*. London, U.K.: Chapman & Hall, 1984.

[5] W. Bridewell, P. W. Langley, L. Todorovski, and S. Džeroski, "Inductive process modeling," *Mach. Learn.*, vol. 71, pp. 1–32, Apr. 2008.

[6] S. L. Brunton, J. L. Proctor, and J. N. Kutz, "Discovering governing equations from data by sparse identification of nonlinear dynamical systems," *Proc. Nat. Acad. Sci. USA*, vol. 113, no. 15, pp. 3932–3937, Apr. 2016.

[7] H. Cao, F. Recknagel, L. Cetin, and B. Zhang, "Process-based simulation library SALMO-OO for lake ecosystems. Part 2: Multi-objective parameter optimization by evolutionary algorithms," *Ecolog. Informat.*, vol. 3, no. 2, pp. 181–190, Apr. 2008.

[8] S. D. Cohen, A. C. Hindmarsh, and P. F. Dubois, "CVODE, a stiff/nonstiff ODE solver in C," *Comput. Phys.*, vol. 10, no. 2, p. 138, 1996.

[9] S. Džeroski and L. Todorovski, "Equation discovery for systems biology: Finding the structure and dynamics of biological networks from time course data," *Current Opinion Biotechnol.*, vol. 19, no. 4, pp. 360–368, Aug. 2008.

[10] J. J. Durillo and A. J. Nebro, "JMetal: A Java framework for multi-objective optimization," *Adv. Eng. Softw.*, vol. 42, no. 10, pp. 760–771, Oct. 2011.

[11] S. Dzeroski and L. Todorovski, "Discovering dynamics: From inductive logic programming to machine discovery," *J. Intell. Inf. Syst.*, vol. 4, no. 1, pp. 89–108, Jan. 1995.

[12] L. Ferariu and A. Patelli, "Multiobjective genetic programming for nonlinear system identification," in *Adaptive and Natural Computing Algorithms*. Berlin, Germany: Springer, 2009, pp. 233–242.

[13] S. Ghosh, S. Das, A. V. Vasilakos, and K. Suresh, "On convergence of differential evolution over a class of continuous functions with unique global optimum," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 42, no. 1, pp. 107–124, Feb. 2012.

[14] F. Giri and E.-W. Bai, Eds., *Block-oriented Nonlinear System Identification* (Lecture Notes in Control and Information Sciences), vol. 404. Berlin, Germany: Springer, 2010.

[15] G. J. Gray, D. J. Murray-Smith, Y. Li, K. C. Sharman, and T. Weinbrenner, "Nonlinear model structure identification using genetic programming," *Control Eng. Pract.*, vol. 6, no. 11, pp. 1341–1352, Nov. 1998.

[16] G. Kerschen, K. Worden, A. F. Vakakis, and J.-C. Golinval, "Past, present and future of nonlinear system identification in structural dynamics," *Mech. Syst. Signal Process.*, vol. 20, no. 3, pp. 505–592, Apr. 2006.

[17] J. R. Koza, *Genetic Programming II: Automatic Discovery of Reusable Programs*. Cambridge, MA, USA: MIT Press, 1994.

[18] P. W. Langley, H. A. Simon, G. Bradshaw, and J. M. Zytkow, *Scientific Discovery: Computational Explorations of the Creative Processes*. Cambridge, MA, USA: MIT Press, 1987.

[19] L. Ljung, *System Identification: Theory for the User*. Upper Saddle River, NJ, USA: Prentice-Hall, 1999.

[20] A. Sage, "Perspectives on system identification," *IFAC Proc. Volumes*, vol. 17, no. 2, pp. 597–601, Jul. 1984.

[21] J. Madar, J. Abonyi, and F. Szeifert, "Genetic programming for system identification," in *Proc. 4th Int. Conf. Intell. Syst. Design Appl. (ISDA)*, Budapest, Hungary, 2004, pp. 43–48.

[22] R. I. Mckay, N. X. Hoai, P. A. Whigham, Y. Shan, and M. O'Neill, "Grammar-based genetic programming: A survey," *Genetic Program. Evolvable Mach.*, vol. 11, nos. 3–4, pp. 365–396, Sep. 2010.

[23] K. Rodriguez-Vazquez, C. Fonseca, and P. Fleming, "Identifying the structure of nonlinear dynamic systems using multiobjective genetic programming," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 34, no. 4, pp. 531–545, Jul. 2004.

[24] M. Schmidt and H. Lipson, "Distilling free-form natural laws from experimental data," *Science*, vol. 324, no. 5923, pp. 81–85, Apr. 2009.

[25] J. Schoukens, R. Pintelon, Y. Rolain, M. Schoukens, K. Tiels, L. Vanbeylen, A. Van Mulders, and G. Vandersteen, "Structure discrimination in block-oriented models using linear approximations: A theoretic framework," *Automatica*, vol. 53, pp. 225–234, Mar. 2015.

[26] J. Schoukens, M. Vaes, A. Esfahani, and R. Relan, "Challenges in the identification of discrete time block oriented models for continuous time nonlinear systems," *IFAC-PapersOnLine*, vol. 48, no. 28, p. 596–601, 2015.

[27] M. Schoukens and K. Tiels, "Identification of block-oriented nonlinear systems starting from linear approximations: A survey," *Automatica*, vol. 85, pp. 272–292, Nov. 2017.

[28] M. Schoukens, K. Tiels, M. Ishteva, and J. Schoukens, "Identification of parallel wiener-hammerstein systems with a decoupled static nonlinearity," *IFAC Proc. Volumes*, vol. 47, no. 3, pp. 505–510, 2014.

[29] J. Tanevski, L. Todorovski, and S. Džeroski, "Learning stochastic process-based models of dynamical systems from knowledge and data," *BMC Syst. Biol.*, vol. 10, no. 1, p. 30, 2016.

[30] K. Tashkova, J. Šilc, N. Atanasova, and S. Džeroski, "Parameter estimation in a nonlinear dynamic model of an aquatic ecosystem with meta-heuristic optimization," *Ecolog. Model.*, vol. 226, pp. 36–61, Feb. 2012.

[31] L. Todorovski, W. Bridewell, O. Shiran, and P. W. Langley, "Inducing hierarchical process models in dynamic domains," in *Proc. 20th Nat. Conf. Artif. Intell. (NCAI)*, K. S. Veloso, Ed. Pittsburgh, PA, USA: AAAI Press, 2005, pp. 892–897.

[32] L. Todorovski and S. Džeroski, "Integrating domain knowledge in equation discovery," in *Computational Discovery of Scientific Knowledge* (Lecture Notes in Computer Science), vol. 4660, L. Todorovski and S. Džeroski, Eds. Berlin, Germany: Springer, 2007, pp. 69–97.

[33] D. Čerepnalkoski, K. Taškova, L. Todorovski, N. Atanasova, and S. Džeroski, "The influence of parameter fitting methods on model structure selection in automated modeling of aquatic ecosystems," *Ecolog. Model.*, vol. 245, pp. 136–165, Oct. 2012.

[34] P. Whigham and F. Recknagel, "Predicting chlorophyll–A in freshwater lakes by hybridising process-based models and genetic algorithms," *Ecolog. Model.*, vol. 146, nos. 1–3, pp. 243–251, Dec. 2001.

[35] T. Wigren and J. Schoukens, "Three free data sets for development and benchmarking in nonlinear system identification," in *Proc. Eur. Control Conf. (ECC)*, Jul. 2013, pp. 2933–2938.

[36] Q. Zhang, "Nonlinear system identification with output error model through stabilized simulation," *IFAC Proc. Volumes*, vol. 37, no. 13, pp. 501–506, Sep. 2004.

LJUPČO TODOROVSKI is currently a Full Professor of informatics with the Faculty of Public Administration, University of Ljubljana, and also a Senior Researcher with the Department of Knowledge Technologies, Jožef Stefan Institute, Ljubljana, Slovenia. His research interest is in the field of machine learning. He develops algorithms for learning models of dynamical systems from observations of their behavior and domain-specific knowledge. In collaboration with researchers from various scientific domains he applies computational, machine learning and data mining methods to practical problems in earth sciences, biomedicine, and administrative sciences.

JUŠ KOCIJAN (Senior Member, IEEE) received the Ph.D. degree in electrical engineering from the Faculty of Electrical Engineering, University of Ljubljana, Ljubljana, Slovenia. He is currently a Senior Researcher with the Jožef Stefan Institute, Ljubljana, and a Professor of electrical engineering with the University of Nova Gorica, Nova Gorica, Slovenia. His research interest includes the modeling and control of dynamic systems. His other activities include serving as an editor and on the editorial boards of research journals, serving as a member for the IFAC Technical Committee on computational intelligence in control. He is a Senior Member of the IEEE Control Systems Society, and a Member of SLOSIM-Slovenian Society for Simulation and Modeling and Automatic Control Society of Slovenia.

NIKOLA SIMIDJIEVSKI received the Ph.D. degree from the Jožef Stefan International Postgraduate School, Slovenia, in 2016, on the topic of learning ensembles of process-based models of dynamic systems. He is currently a Postdoctoral Research Fellow with the Department of Computer Science and Technology, University of Cambridge, U.K. He is also a Visiting Researcher with the Jožef Stefan Institute, Slovenia. His research interests are in the field of machine learning and data mining, i.e., development of novel machine learning and data mining algorithms and their application in life and earth sciences. He has participated in several national and international projects including AI for Integrated Cancer Medicine, MAESTRA (Learning from Massive, Incompletely annotated, and Structured Data) and The HBP (The Human Brain Project). He has been a member of the PC of many conferences (such as ICML, ECML PKDD, and others) and served as a Reviewer for many journals including IEEE Access.

SAŠO DŽEROSKI received the Ph.D. degree in computer science from the University of Ljubljana, in 1995. He is currently a Senior Researcher with the Department of Knowledge Technologies, Jožef Stefan Institute, Ljubljana, Slovenia, and a Professor with the Jožef Stefan International Postgraduate School, Ljubljana. His research interests fall in the field of artificial intelligence (AI), focusing on the development of data mining and machine learning methods for a variety of tasks — including the prediction of structured outputs and the automated modeling of dynamic systems — and their applications to practical problems from science and engineering, e.g., environmental sciences (ecology) and life sciences (biomedicine). In 2008, he was an Elected Fellow of the European AI Society for his Pioneering Work in the field of AI and Outstanding Service for the European AI community. In 2015, he became a Foreign Member of the Macedonian Academy of Sciences and Arts. In 2016, he was elected as a member of Academia Europaea.

• • •