Recent Advancements & Methodologies in System Identification: A Review

Ihsan Mohd Yassin, Mohd Nasir Taib and Ramli Adnan
Faculty of Electrical Engineering
Universiti Teknologi Mara (UiTM)
Shah Alam, MALAYSIA
ihsan.yassin@gmail.com

***Abstract*- System Identification (SI) is a discipline in control engineering concerned with inferring mathematical models from dynamic systems based on its input/output observations. Rich literature is available regarding SI due to its applications in understanding complex systems as well as to design control systems for complex processes. A summary of those literatures is presented in this paper, which covers general classifications of SI, methodologies for design and implementation of the models, as well as recent advancements in application of optimization techniques for SI. It is hoped that this paper would serve as a guide for budding practitioners regarding the fundamentals of the discipline, while serving as a summary of new research trends for experienced researchers in the area.**

***Index Terms*—System Identification, System Identification Methodology, Optimization Applications in System Identification**

I. Introduction

System Identification (SI) is a control engineering discipline concerned with the inference of mathematical models from dynamic systems based on their input and/or output measurements [1-5]. It is fundamental for system control, analysis and design [2], where the resulting representation of the system can be used for understanding the properties of the system [1] as well as forecasting the system's future behavior under given inputs and/or outputs [1, 6].

SI is a significant research area in the field of control and modeling due to its ability to represent and quantify variable interactions in complex systems. Several applications of SI in literature are for understanding of complex natural phenomena [7-11], model-based design of control engineering applications [8, 12-16], and project monitoring and planning [10, 11, 17-19].

This paper is organized as follows: an introduction to SI and its broad classifications are presented in Section II. This is followed by the SI methodology in Section III. The section describes each process in SI, from data collection and treatment, selection of model type and estimators, structure selection and parameter estimation, and validation methods. Next, the applications of two established optimization algorithms (Genetic Algorithm (GA) and Particle Swarm Optimization (PSO)) for SI are presented in Section 0. Finally, concluding remarks are presented in Section V.

II. Overview of SI

This section presents a review of the applications of SI in Section A, followed by distinctions between parametric/non-parametric and linear/non-linear modeling methods in Section B and Section C, respectively.

A. Applications of SI

Based on literature, the applications of SI are broadly classified into the following categories:

1. Understanding of complex natural systems.
2. Model-based design and implementation of control systems.
3. Planning and monitoring of engineering projects.

Natural systems are complex as many factors affect their behavior, and the behavior of some of these factors is only partially known [8]. Examples of applications in this area are rainfall-runoff prediction of river basin [10], modeling of the cardiovascular system [9], weather modeling for placement optimization of monitoring stations [11], and human joint modeling during movement [7].

Many engineering applications require the construction of models as part of control system design [8]. This is because adaptive controllers build on SI models have increased robustness in the presence of unknown model parameters and external disturbances [12]. In [13], SI was performed a mechatronic maneuvering system for marine vessels. The authors suggested that an accurate representation of ship dynamics may help in model-based designs such as controller design and analysis. Similarly, works by [14] presented a controller for roll angle and path of motorcycles based on SI performed on measured data. Another example in [20] used SI to model noise in a vehicle steering controller to improve the design of a Model Predictive Control (MPC) controller. In [15], SI was used to estimate the power consumption of Central Processing Units (CPUs) to develop an intelligent CPU power management system controller. In [16], identification work of engine controllers has led to reduced requirements for experiments and trials for the design process.

Another application of SI is to assist in planning and monitoring of engineering projects. Several examples in this category are placement of weather monitoring stations based on SI results [11], monitoring of conditions in civil structures [17], identification of rainfall-runoff model for operational planning of hydroelectric power plants [10], prediction of gas content in coal beds [18], and identification of power system characteristics for reliable planning and operation [19].

B. Parametric vs. Non-Parametric Modelling

The task of SI can be divided into two categories based on the availability of knowledge regarding the structure of the model. If the model structure is assumed to be known prior to identification, the identification task is considered to be parametric [5, 21-24]. The identification problem is therefore reduced to a parameter estimation task [21-23, 25]. Examples of parametric identification methods are state-space, regression, and subspace identification [22]. Parametric methods are more robust to noise [22], require less data to construct the model [22], as well as improved parsimony and accuracy [26]. However, parametric methods require an insight of the system prior to modeling, information that may not be readily available.

A non-parametric identification method makes minimal or no prior assumptions of the model structure [21-24]. This approach makes the modeling task significantly more complex compared to parametric identification, as the model class options are extremely rich [27] and there are more unknown that need to be considered in extension to the parameter estimation task [21, 22]. Therefore, this type of method requires more data compared to its counterpart [22]. Examples of non-parametric identification methods are polynomials, Fourier series, orthonormal functions and kernel estimators [21]. However, the non-parametric approach is more practical and flexible in cases where the system's physical insight is unavailable.

The concept of parametric and non-parametric methods translates into three different types of modeling techniques: white box, grey box or black box. The white box modeling technique constructs the model based on physical equations that govern the system process [28]. This type of modeling technique requires extensive knowledge of the system behavior [28], which is closely related to parametric modeling.

The black box modeling technique constructs the model based only on the data acquired from the system, and does not rely on other information about the system [28]. No initial assumptions about the model structure are made, rather the modeling technique's concern is to create a generic model that maps the input-output relationship of the dataset [29, 30]. Black box models have the advantage of practicality and flexibility, as structural information about the system may not be readily available. However, a disadvantage is that the model complexity increases [31]. Another advantage is black box identification can model dynamics that are insufficiently captured by mathematical models [8, 32].

The grey box modeling technique is a middle ground between black box and white box modeling. Black box estimation is performed supported by a priori knowledge about the system [28, 29]. This knowledge (such as steady-state information) is included as auxiliary information to assist in the construction of the system [28]. This modeling technique reduces the complexity of black box model construction using physical insights about the nature of the system [31].

C. Linear vs. nonlinear

Two types of modeling techniques are usually employed in SI: linear or non-linear. The choice between linear and nonlinear models is a tradeoff between complexity and accuracy of the model. However, because all real systems are inherently nonlinear [1, 5, 12, 33-37], nonlinear identification is more advantageous than linear identification in terms of model representativeness [38-43].

Linear identification assumes a linear relationship between the data, and treats the nonlinearity as part of the uncertainty [5, 21, 42, 44]. Linear identification is suitable when the system exhibits weak nonlinearities [34]. Examples of application of linear models are available in [3, 7, 30, 45-48]. Linear models performance is reduced significantly if the system exhibits strong nonlinear behavior (presence of nonlinearity, time-varying, and delay characteristics) [21, 34, 37]. To minimize the effect of nonlinearities, several works have shown some success with linearizing a nonlinear system around a desired operating point or decomposing the nonlinear system into smaller piecewise linear models. Examples of this method are [5, 46, 49]. However, the performance of these type of models decrease in systems with strongly nonlinear behavior or possessing wide operating ranges [5].

In cases where the system is strongly nonlinear, linear model performance is unsuitable, and nonlinear SI models are preferable [1, 34, 35, 37]. Nonlinear models incorporate the nonlinearities of the system into the model. A review of nonlinear models has been presented in [23]. Nonlinear identification has become more acceptable due to the following factors [38]: (i) theoretical advancements of nonlinear systems, which has provided design methodologies to model nonlinear systems; (ii) development of efficient identification methods for treatment of empirical nonlinear models; (iii) improvements in control hardware and software that enables the incorporation of complex nonlinear models in its design. Compared with linear identification (where the works are mature and well-documented), nonlinear SI modeling is the subject of active research [5], and a review of papers submitted in this area shows an increasing trend [50]. However, the use of nonlinear models obscures insight into the true system model as there are few analysis tools available to analyze the models [5].

III. SI PROCEDURE

It is desirable for the model to be as representative as possible with respect to the actual system being modeled [4]. Therefore, the SI process consists of structured methods to achieve this.

The SI procedure can be broadly sub-divided into three steps, name model order/structure selection (Section E), parameter estimation (Section F) and model validation (Section G) [1, 51]. However, in this review, three additional sub-divisions were also considered worthy of treatment, which are data collection (Section A), data preprocessing (Section B), model selection (Section C) and model estimator selection (Section D).

A. Data Collection

Data collection is defined as the task of collecting input-output data from the system for the purpose of identification. It is an important step as SI relies heavily on the quality of data being collected from the system. Two types of data collection methods were found in literature: 1) collection of data from historical or observational data, 2) collection of data by exciting the system with a predefined input.

In certain circumstances where observational data is already available for identification, the input-output pairs are already defined and there is no requirement for the collection of data from designed input signals. Examples of this method are [11, 52, 53]. Both papers performed identification on readily-available historical observational data.

In the second case, the data collection procedure is based on collecting the signal produced at the output under exciting conditions given by controlled input signals [34]. The choice of input signals depends on the method chosen for identification of the system [34]. Furthermore, the probing signal should engineered such that it is within frequency band of interest [19].

Various excitation signals have been used as inputs for data collection of different types of systems. The step response is used to investigate the stability of power systems because it represents different frequency excitations [54]. However, if the system's transfer function is desired, it can be revealed using the impulse signal [54].

Pseudo-Random Binary Sequence (PRBS) have been used as inputs in systems that have binary switching modes [55-57]. In cases where the system is excited by multiple signal levels, reference [58] used the Pseudo-Random Multilevel Sequence (PRMS) as inputs.

Other researchers have also reported using sine signals for data collection. Several examples are swept-sine input signals [34, 59] and low-level multi-sine probing signals under focused frequency ranges [19]. The application of the sine cardinal (sinc) input signal has also been presented in modeling of a synchronous generators [54].

Sampling the output under different frequency ranges and operating conditions have also been reported. In [60], a combination of inter-sample response and multi-rate sampling was used to collect data for a discrete-time Iterative Learning Control (ILC) algorithm. Similarly, reference [28, 61] collected response signals under various operational and excitation conditions. In [17], the Natural Excitation Technique (NExT) was applied to extract data characteristics under different configurations. Reference [25] reported the collection of open-loop data from a Pulse Width Modulation (PWM) Direct Current to Direct Current (DC-DC) converter for designing its closed-loop controller. In [57], data collection was performed under three bandwidth-limited operating conditions.

In certain linear identification applications, the input signal is formulated to suppress nonlinear characteristics of the system. An example of this input signal design is shown in [57], where the magnitude of the input signals were made sufficiently small to minimize nonlinear effects of the system, while being large enough to minimize the effect of noise.

B. Preprocessing and Data Division

Preprocessing is defined as the task of transforming raw data into a more complete and/or representative form for the purpose of identification. The task is important because of several reasons:

1. To ensure that the data is within an acceptable range.
2. To remove unnecessary trends/noise that may affect identification results.
3. To complete missing points from the dataset.
4. To adjust the size of the sampled data.

Magnitude scaling is used to adjust the magnitude of the data to within an acceptable range. Several papers that utilize this preprocessing method are [61-63]. Several authors set the range according to arbitrary values [62, 63], although ANN practitioners tend to set it to between -1 and +1 [61]. Magnitude scaling is particularly useful in Multi-Layer Perceptron (MLP)-based SI estimators. The MLP structure for function approximation typically utilizes the sigmoid activation function in its hidden layer(s) [64]. The sigmoid activation function has a squashing effect on inputs beyond the range of -1 and +1 [64]. If the inputs are not scaled, the weight vectors in the MLP tend to be too small or too large, which would affect the training process [61].

Filtering is used to remove noise from the dataset. Methods found in literature used noise filtering algorithm [65], time-series averaging [20, 25], magnitude-based filtering [66] or frequency filters [9, 20, 67] to remove unwanted frequency components. Additionally, data normalization (removal of mean and standardization of the variance) can be used to remove trends in the data. An example of this method is shown in [68, 69].

Interpolation can be used if data is missing or insufficient. The method involves estimating missing data points based on available data. An example of this method was found in [10], where missing data from unknown rainfall gauges were synthesized using interpolation to construct a rain runoff model.

Resampling of signals is typically used to reduce the number of data points to a more manageable value or to remove noise. Examples of research that have used this method to reduce data size are [9, 67]. In [25], the sampling interval was adjusted so that sampling takes place after the switching oscillations have decayed in order to reduce the noise.

It is preferable to separate between training and testing datasets. A model should have good accuracy not only over the training dataset, but also on the independent testing set [1]. The training dataset is used for updating the parameters of the model. Therefore, the model will naturally fit this data well. The testing dataset is important because it serves as an independent measure to evaluate the model's generalization ability. The division of data can be performed using methods such as block, interleaving, or random division [55, 63, 70-72]. Block division divides the dataset in blocks according to a predetermined ratio. Interleaving divides the dataset according to the position of the data in the dataset. Finally, random sampling divides the dataset in no particular order. Another approach found in literature constructs training and testing datasets from independently-performed experiments [20].

C. Model Selection

Model selection involves the determination of a suitable model to represent the data. A rich choice of models is available for SI: Volterra Series Expansion (VS), Hammerstein Model (HM), Wiener Model (WM), Non-linear Auto-Regressive Moving Average Model with Exogenous Inputs (NARMAX) and its derivatives, and application-specific models. They are described in Section 1) to Section 5).

1) VS

VS is a non-parametric [27, 50] nonlinear SI model that takes the form of [73]:

$$y(t) = y_1(t) + y_2(t) + y_3(t) + \cdots \tag{1}$$

where:

$$y_1(t) = \int_{-\infty}^{+\infty} h_1(\tau)x(t-\tau)\ d\tau \tag{2}$$

$$y_2(t) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} h_2(\tau_1,\tau_2)x(t-\tau_1)x(t-\tau_2)\ d\tau_1\ d\tau_2 \tag{3}$$

$$y_3(t) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} h_3(\tau_1,\tau_2,\tau_3)\,x(t-\tau_1)x(t-\tau_2)x(t-\tau_3)\ d\tau_1\ d\tau_2\ d\tau_3 \tag{4}$$

where functions $h_1(\tau)$, $h_2(\tau_1,\tau_2)$, $h_2(\tau_1,\tau_2,\tau_3)$ and $h_2(\tau_1,\tau_2,\tau_3,\dots\tau_n)$ are known as Volterra kernels - generalizations of the linear impulse response function. The form of VS is such that the system output at a particular time, $y(t)$, is equivalent to the infinite sum of multidimensional convolutional integrals of its past inputs. The VS identification model therefore, is able to retain past memory in order to predict its future outputs. The applicability of VS for modeling time-series data is based on the Frechet Approximation Theorem, which states that under some assumptions, any time-invariant can be uniformly approximated with arbitrary precision by a sufficiently high finite order VS [74]. The VS kernels are obtainable by minimizing a cost function (such as the Mean Squared Error (MSE)) with respect to the VS coefficients. [36].

Advantages of VS are its structural generality and versatile modeling abilities [75], good approximation for low-order polynomial models [27], linear-in-the-parameter nature (due to its polynomial representation) [35, 76], and robustness towards output noise [76]. However, criticisms towards the VS are directed towards its low effectiveness in approximating high-degree or non-polynomial type nonlinearities [27], high estimation error and slow convergence in cases where the problem is ill-conditioned [35], sensitivity towards noise in its input [76], large number of parameters for [29, 35, 36] leading to high computational cost [35]. Furthermore, VS does not have memory of the output, which makes it unsuitable for modeling systems where the output is also dependent on its past values.

2) HM

HM is a cascaded nonlinear identification model with a static (zero-memory) nonlinear element followed by a time-varying linear element [48, 67, 77-79], as shown in Figure 1.

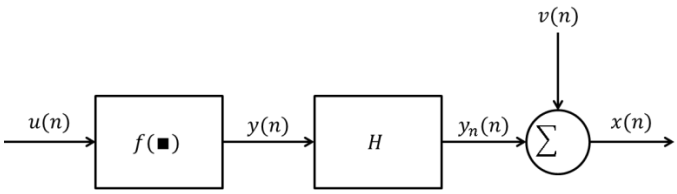

Figure 1: HM with measurement noise

Its general representation is deliberated here [80]. The nonlinear HM block is defined as:

$$\begin{aligned} y(n) &= f(u(n)) \\ &= c_1u(n) + c_2u^2(n) + \cdots + c_{nc}u^{nc}(n) \end{aligned} \tag{5}$$

and the model output is characterized by:

$$x(n) = y_n(n) + v(n) \tag{6}$$

where $v(n)$ is zero-mean additive white Gaussian noise. Block $y_n(n)$ is characterized by the following relationship:

$$\begin{aligned} y_n(n) &= H(z^{-1})y(n) \\ &= \frac{B(z^{-1})}{A(z^{-1})}y(n) \end{aligned} \tag{7}$$

Both $A(z^{-1})$ and $B(z^{-1})$ are polynomials [34, 36, 80] in the form of:

$$A(z^{-1}) = 1 + a_1z^{-1} + a_2z^{-2} + \cdots + a_{na}z^{-na} \tag{8}$$

$$B(z^{-1}) = 1 + b_1z^{-1} + b_2z^{-2} + \cdots + b_{nb}z^{-nb} \tag{9}$$

where $z^{-1}y(n) = y(n-1), z^{-2}y(n) = y(n-2), \dots$ and so forth. $a_1, a_2, a_3, \dots, a_{na}$ and $b_1, b_2, b_3, \dots, b_{nb}$ are estimated parameters of the HM such that they minimize the difference between the approximated model and actual observations [80]. This task can be achieved using methods such as extended LS [79]. Methods such as Auto-Regressive Model with Exogenous Inputs (ARX) [48], Frequency Impulse Response (FIR) [77]

and Auto-Regressive Moving Average Model with Exogenous Inputs (ARMAX) [79] has been used to describe the linear block.

The HM is appealing because its block-oriented structure is suitable for many system processes [50, 77]. However, some weaknesses of the HM model are non-sensitivity to asymmetric dead zone nonlinearities in the control inputs [78], and difficulty to construct with insufficient prior information [78]. For highly nonlinear systems, the static nonlinear block cannot be expressed as a finite-order polynomial, resulting in a high-dimensional linear block [77]. Furthermore, with the structure of the model already assumed, HM becomes a parametric identification model thus inheriting method's weaknesses.

3) WM

WM is a cascaded nonlinear identification model with a time-varying linear block followed by a static (zero-memory) nonlinear element [50, 67, 81, 82], as shown in Figure 2. It is a structurally-reversed version of HM [83], where the dynamics of the system is described by a linear model and there is static nonlinearity at the output [67].

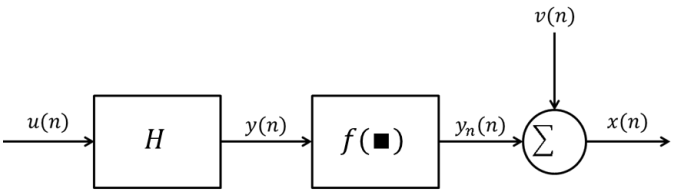

Figure 2: WM with measurement noise

The output of the linear block, $y(n)$, is given by:

$$\begin{aligned} y(n) &= H(z^{-1})u(n) \\ &= \frac{B(z^{-1})}{A(z^{-1})}u(n) \end{aligned} \quad (10)$$

where $A(z^{-1})$ and $B(z^{-1})$ are similar to Eq. (8) and Eq. (9) from Section 2).

The output of WM, $x(n)$, is given by:

$$x(n) = y_n(n) + v(n) \quad (11)$$

and the output of the nonlinear block $y_n(n)$ is given by:

$$\begin{aligned} y_n(n) &= f(y(n)) \\ &= c_1 y(n) + c_2 y^2(n) + \cdots + c_{nc} y^{nc}(n) \end{aligned} \quad (12)$$

$y_n(n)$ is expressed in the form where $y(n)$ is orthogonalized using methods such as Gram-Schmidt [36]. This step helps to construct closed-form solutions and kernel estimation separation [36]. The nonlinear block is typically unknown and can be represented by polynomials among other methods [67].

Several combinations of linear-nonlinear block representation methods were found in literature. In [81], the linear subsystem was represented using probabilistic distribution while the nonlinear subsystem was modeled using Kalman filtering. Work by [82] used a linear FIR filter with a nonlinear polynomial representation.

WM is appealing because it is a nonlinear extension to the already-established linear modeling techniques [67]. However, some weaknesses of the model are it inherits weaknesses from parametric modeling due to structure assumed to be known. Additionally, both HM and WM assume an essentially linear dynamic model with static nonlinearity, which poses a problem when the nonlinearity in the data is dynamic in nature.

4) The Nonlinear Auto-Regressive Model with Exogenous Inputs (NARMAX) and Derivatives

The NARMAX [84] and its derivatives (NARX (NARX) and NARMA (NARMA)) are nonlinear non-parametric identification models for SI [4, 24, 85, 86]. They are powerful, efficient and unified representations of a variety of nonlinear systems and SI models [1, 39, 62, 86-93]. Rich literature is available regarding its success in various electrical, mechanical, medical and natural applications. For examples, see [28, 61, 62, 71].

The NARMAX model outputs is dependent on its past inputs, outputs and residuals [28, 86, 94, 95], while NARX removes the residual terms [61, 94, 96-99] and NARMA removes the input terms from the model. The NARX model is a generalized version of the Auto-Regressive with Exogenous Inputs (ARX) [24] estimated in a non-recursive manner. The NARX/NARMA/NARMAX model can be constructed using various methods, such as polynomials [52, 67, 100, 101], Multilayer Perceptrons (MLP) [55, 56, 69, 91], and Wavelet ANNs (WNN) [102, 103], although the polynomial approach is the only method that can explicitly define the relationship between the input/output data.

The NARMAX model takes the form of [52]:

$$\begin{aligned} y(t) = f^d[&y(t-1), y(t-2), \dots, y(t-n_y), \\ &u(t-n_k), u(t-n_k-1), \dots, u(t-n_k-n_u), \\ &\varepsilon(t-1), \varepsilon(t-2), \dots, \varepsilon(t-n_\varepsilon)] + \varepsilon(t) \end{aligned} \quad (13)$$

where f^d is the estimated model, $y(t-1), y(t-2), \dots y(t-n_y)$ are lagged output terms, $u(t-n_k), u(t-n_k-1), \dots u(t-n_k-n_u)$ are current and/or lagged input terms, $\varepsilon(t-1), \varepsilon(t-2), \dots \varepsilon(t-n_\varepsilon)$ are lagged residual terms, and $\varepsilon(t)$ are the white noise residuals. Parameter n_k is the input signal time delay, its value is usually 1 except for cases where the input $u(t)$ is required for identification (in which case, $n_k = 0$) [52]. The lagged residuals terms $\varepsilon(t-1), \varepsilon(t-2), \dots \varepsilon(t-n_\varepsilon)$ are obtained recursively after the initial model (based on the input and output terms) is found.

The NARMA model is a time-series derivative of the NARMAX algorithm. Input terms are absent from the NARMA model, therefore the equation becomes:

$$\begin{aligned} y(t) = f^d[&y(t-1), y(t-2), \dots, y(t-n_y), \\ &\varepsilon(t-1), \varepsilon(t-2), \dots, \varepsilon(t-n_\varepsilon)] + \varepsilon(t) \end{aligned} \quad (14)$$

Similar to NARMAX, the lagged residuals terms are obtained recursively after the initial model is found.

NARX is the non-recursive version of the NARMAX model, as it does not reuse residual terms back into the model:

$$y(t) = f^{d}[y(t-1), y(t-2), \dots, y(t-n_y), u(t-n_k), u(t-n_k-1), \dots, u(t-n_k-n_u)] \quad (15)$$

Similar to the NARMAX model, n_k is typically 1 but may be changed to 0 if input $u(t)$ is required to define the model. The lagged input, output and residual terms can also be multiplied with each other to model higher-order dynamics beyond single-term polynomials.

The identification method for NARMAX and its derivatives are performed in three steps [95]. Structure selection is performed to detect the underlying structure of a dataset. This is followed by parameter estimation to optimize some objective function (typically involving the difference between the identified model and the actual dataset) [61]. The structure selection and parameter estimation steps are typically performed using the Error Reduction Ratio (ERR) method [95]. In real-life cases where bias may arise from correlated or colored noise, the more powerful NARMAX model can be used [98]. The NARMAX model recursively adds residual terms to the NARX model to eliminate the bias and improve the model prediction capability [50, 94, 98, 104]. Structure selection and parameter estimation for NARX are done once, while for NARMA and NARMAX, the steps are recursively repeated on the residual set until a satisfactory model is obtained [86]. Finally, the model is validated to ensure that it is acceptable.

A major advantage of NARMAX and its derivatives is that it provides physically interpretable results that can be compared directly, thus providing a way to validate and enhance analytical models where first principle models lack the completeness because of assumptions and omissions during derivations [95]. Furthermore, among all the models studied, it is the only model that embeds the dynamics of nonlinearities directly into the model [66]. Other advantages include model representativeness [61, 62, 85, 96], flexible model structure [85], stability (NARX) [4], reliability over a broad range of applications [1, 93], as well as reasonable parameter count and computational costs (for reasonably-sized model structures)[61, 62, 105].

Several weaknesses are listed here: In cases where the system is highly nonlinear or model structure is large, the NARMAX model and its derivatives suffer from complex model structures [85, 92, 93, 105] leading to difficult transition to control system design [93]. This problem is compounded by the difficulty of estimating the parameters for the Moving Average (MA) part of the model, which is computationally inconvenient and unreliable [45] as the residual regressors are usually numerically ill-conditioned. However, if a simpler NARX model is selected, it may introduce bias in the absence of a disturbance model [94]. Furthermore, in creating the NARMAX model, the residuals from the NARX identification is carried forward leading to accumulation of error in the model [62].

5) Application-Specific Models

Apart from the general models presented in Section 1) to Section 4), there are other SI models used for specific application domains. In [67], three application-specific models (Virtual Muscle Model, Bobet and Stein Modeling of Force Production, and Bartha and Thompson Modeling of Force Production) were used to model the isometric force of a primate's eye muscles in response to electrical stimulation. The Error-in-Variable Model (EIVM) was used to model noise in systems [106], while the Mesoscale Weather Model was used to model thermal capacities for monitoring station placement in a Dynamic Thermal Rating application [11]. Works by [107] performed SI and control of a tendon-based robotic actuation system using Static Properties Modeling, Viscoelastic Model for the Tendon, Model of the Tendon Friction and Lumped-Parameter Tendon Model. While these models are superior in their application domains, they cannot be used for modeling general systems.

D. Model Estimator Selection

Once the choice of model type has been made, the type of estimator has to be chosen. An estimator may be pre-selected based on physical insight, prior knowledge or experience [26]. Alternatively, the model estimator may be chosen from a set of candidate estimators that produces the optimal model. A review of several model estimators is presented in Section 1) to Section 7).

1) Polynomial Function

Polynomial SI estimators represent the SI model in terms of polynomial basis functions. Polynomial of model estimator is the most common SI model [105], and have been used extensively in several types of models structures, namely NARMAX/NARX [85, 86, 97, 104], HM and WM [36] and VS [36] model types. Several applications of the polynomial estimator can be found in [20, 28, 57, 85, 95, 105, 108, 109].

An advantage of the polynomial model estimator is its applicability to accurately and uniformly estimate a broad class of continuous nonlinear systems [36, 105]. Its Linear-in-the-Parameter (LITP) nature is well-structured for adaptive learning with provable learning and convergence conditions [1]. This property is appealing across all traditional engineering sectors for monitoring, control and supervision [1].Furthermore, LITP allows parameters to be easily determined using linear estimation methods [70, 85, 94, 97, 110].

A disadvantage of the polynomial estimator is the availability of a large number of potential candidate structures. The structure selection step is required to select an optimal set of regressors from the large number of potential candidates [94]. Unsatisfactory results are obtained if the candidate structure is under-parameterized, while over-parameterization leads to an excessively large number of terms and poor estimation due to numerical ill-conditioning [94, 111].

2) Artificial Neural Network (ANN)

ANNs are defined as a massively parallel distributed processor consisting of interconnected simple processing units that has the ability to store experiential knowledge through a learning process [4, 64, 112]. ANN is a widely used function estimator due to its various appealing properties: universal approximation ability [61, 77, 99, 113-115], strong representation of unknowns and complex nonlinearities [8, 18, 37, 58, 61, 93, 116], simple implementation [112], and flexible black-box design methodology [18, 61, 71, 96, 110].

Despite its many advantages, several limitations of the ANN are excessive amounts of parameters [28, 29, 68], difficulty to determine optimal network parameters [58, 99], computationally-expensive and time-consuming training process [58, 68, 115, 117, 118], lack of model transparency [28, 68], sensitivity to initial weight values [58], over-generalization or under-generalization [61, 117], and tendency to be trapped in local minimas [58, 117].

ANN is divided into many subtypes, namely MLP (MLP), Radial Basis Function ANN (RBFNN), Wavelet ANN (WNN) and Neuro-Fuzzy Models (NFM). Each of these subtypes is described in Section 3) to Section 6).

3) Multi-Layer Perceptrons

It is difficult to separate the topics of ANN and MLP because both terms are often used interchangeably in literature, although they are actually different. Consequently, the author has taken the liberty to redefine and reorganize the ANN/MLP terms based on the design methods described by the research papers reviewed. According to [64], the MLP has several distinguishing characteristics that separate it from other ANN subtypes [64]:

1. The MLP contains one or more layers of hidden units that are not part the input/output of the network. The hidden units enable the MLP to learn complex tasks and meaningful features from the input/output relationship.
2. Presence of nonlinear activation functions in its hidden layer.
3. High degree of connectivity between the MLP layers, determined by the weights of the network.

Several parameters can be adjusted to optimize the MLP's predictive capabilities. The structure of the MLP is defined prior to training, where the designer specifies the number of layers the network has. In many MLP applications, the three-layer structure (one input layer, one hidden layer, and one output layer) is preferable [61]. This is because the three-layer structure with sufficient hidden units has been proven to be able to approximate any finite continuous function with good degree of accuracy [99, 113]. However, in several applications [32, 113, 116], more hidden layers have been reported, citing higher generalization ability as the main reason [113].

The number of hidden units reflects the approximation ability of the MLP [113]. This is because hidden units are collectively responsible to map the relationship present between the input and output. Too few hidden units result in decreased generalization due to insufficient mapping information [113]. However, if they are too many hidden units, the network structure is suboptimal due to presence of redundant and useless weights [113], as well as over-fitting of unwanted noise components [113] leading to stability and reliability issues [61]. The most common method of determining this property is trial-and-error [99, 110, 113], although other methods (rule-of-thumb) and algorithms (pruning and growing) are available [77, 99, 113, 119].

For SI purposes, the typical choice of hidden and output activation functions is sigmoid (and its variants [77]) and linear, respectively [61, 71, 96, 110]. This is because SI is function approximation task, thus restricting the activation function to the ones mentioned above.

In models where memory is present or desired, they can be represented in the form of lagged regressors [71] or recurrent connections (feedback of outputs back to the MLP [10]) [4, 8, 28, 62]. The recurrent connections create a closed-loop model, where the feedback introduces additional dynamics so that the MLP learns without much assumption of the inherent system structure [62, 116, 120]. Additionally, recurrent structures are suitable for either one-step or multi-step ahead predictions, where the current output is recycled and used to extend the prediction horizon [10, 96]. However, it should be noted the selection of model also influences the selection between static and recurrent networks [96]. This is reported in [96, 116], where a static NARX network has been shown to outperform recurrent MLP structures. Additionally, static MLPs are simpler and more stable in the absence of recurrent connections [4].

The training algorithm are typically gradient-based methods (such as Levenberg-Marquardt (LM) [28, 61, 113], and Newton-Rhapson (NR) method [93]). The superiority of LM has been reported in a comparison of various training algorithms [72]. Apart from gradient-based methods, training algorithms based on evolutionary algorithms have been presented [37, 113], while specific training algorithms exist for specialized MLP types [114]. Decaying weight adjustments and Bayesian regularization have been performed by several authors [32, 61] to improve the convergence of the network while minimizing parameter expansion.

4) Radial Basis Function ANN (RBFNN)

The RBFNN is a three-layer feed-forward ANN with Gaussian activation function in its hidden units and output unit(s) that outputs a linear weighted sum of the hidden units [26, 58, 121-124]. Each hidden unit in the RBFNN outputs a value of the difference between its weight and input from the center of the Gaussian activation function [122]. Based on the Gaussian activation function, when the input is close to the hidden unit's center, the unit output is high. As the distance between the input and the center increases, the output will rapidly decrease to zero [124].

The construction of RBFNN is performed in two stages. The first stage is the determination of basis function centers and widths. This stage is typically performed using randomly-selected inputs, or using clustering, cross-validation or Orthogonal LS and its derivatives (OLS) method [58, 123, 125]. A training algorithm is then employed to simultaneously adjust the hidden layer's center vectors and variances and the

weights to minimize the difference between the actual and predicted response [124]. This stage is typically performed using LS (due to its LITP nature), gradient-descent based algorithms, Expectation-Maximization (EM) or other types of evolutionary algorithms [54, 58, 125, 126].

Among the advantages of RBFNN is fast computational and training speed [58, 124, 127], near-optimal convergence [1, 58], LITP nature [6], structural and design simplicity [58, 122, 127], and transparent learning process [122].

The problem with RBFNN is with regard to determination of basis function centers and widths. Clustering methods are sensitive to initial values, while determination of basis function centers and widths guided by randomly-selected inputs tend to over-fit the data [58]. Additionally, gradient-descent, EM or evolutionary-based methods used in the second construction stage are computationally expensive and have a tendency to be trapped in suboptimal solutions [58].

5) Wavelet ANNs (WNN)

WNN are a type of ANN that utilizes a linear combination of wavelet activation function for the hidden layer instead of the typical Gaussian or sigmoid functions [50, 128]. It belongs to a class of local function-expansion LITP model structures that use the powerful wavelet decomposition technique to represent signal nonlinearities [50, 105]. Much of the WNN approximation ability is derived from its localization in both time and frequency domain [105, 128].

There are generally two structure types of WNN [128]. The first type, continuous wavelet transform is used as the activation function [128]. This type of WNN modifies the dilation and translation parameters together with the output layer weights during training [50, 128]. The second type of WNN uses discrete wavelet transform as the activation function in its hidden layer [128]. The training process for this type of WNN only modifies the output layer weights [128].

WNN has received much attention because they can approximate nonlinearities well, even in signals that have discontinuities [105]. Among the advantages of WNN are flexibility [105], high-precision approximation [105, 128], quick convergence [128] and compact structure [105, 128]. However, the drawbacks of WNN are expanded number of parameters [29, 105], which results in increased computation relative to other simpler estimators [105].

6) Neuro-Fuzzy Models (NFM)

Fuzzy Logic (FL) is a branch of Artificial Intelligence (AI) intended to emulate the natural decision-making ability of humans under imprecise, uncertain or incomplete information. The decision-making ability of FL is derived from a set of heuristic IF-THEN rules [99, 129, 130]. It is therefore well-suited for processes that have ill-defined, imprecise or partially-available information that cannot be represented in a mathematical manner [8, 63, 99, 131]. Other advantages include model transparency and simple interpretation of the rules [62, 65, 131].

FL provides a straightforward method to decompose the modeling task into local subtasks. These subtasks are relatively easier to handle and improves the general model and control design [130]. However, the fixed grid-partitioning design approach of FL suffers from excessively large number of rules, as well as inflexibility in design [65]. Additionally, the design process is difficult with insufficient expert knowledge [62]. NFM has been designed to address these shortcomings [65].

NFM are hybrid systems in which inference properties of FL is used to enhance the learning ability of the ANN [8, 62, 128, 129]. It belongs in a class of universal approximators, therefore it is able to approximate any continuous function with a good degree of accuracy [12, 65, 115, 129]. The FL membership function parameters, rules, number of fuzzy sets/clusters and ANN weights are adjustable parameters to optimize convergence and model structure [10, 65, 68, 129]. Training is generally performed using the back-propagation and least-squares algorithms and its variants [63, 128, 129, 132], although use of evolutionary algorithms have also been reported [10, 62, 99, 131].

Among the many NFM types is the Adaptive Neuro-Fuzzy Inference System (ANFIS), an online self-constructing ANN with a modified Takagi-Sugeno-Kang (TSK) Fuzzy Inference System (FIS) [63]. Additionally, a NARMAX-based Fuzzy ANN (FNN) was presented in [115], where the input vectors were applied to several layers of membership function conversion, application of the fuzzy basis function and evaluation of fuzzy rules. The layers of the FNN had weighted interconnections, which were adjustable to fit the dataset. Another variant of the NFM was found in [128], which combined WNN with TSK FL to become Fuzzy Wavelet ANNs (FWNN). In [120], a combination of Support Vector Regression (SVR) and a recurrent TSK FNN model to incorporate additional parameter-learning abilities during training. In [131], a Mamdani-type ANFIS model was compared with its TSK counterpart, with improved model accuracy and transparency.

Although proven as good function approximators, some criticisms have been directed towards increased complexity [62] and high parameter count [115, 120, 130], thus resulting in increased computational cost [115]. This flaw has been subject of active research [63, 115, 120]. Furthermore, the embedded ANN presents a black box-type estimator, where the inherent model structure is unclear [62].

7) Other Estimator Types

Apart from model estimators shown in Section 1) and Section 2), there are other less-known model estimator types found in literature. These estimators are presented in this section.

In [9], Dual Laguerre Basis Functions linear approximator were used to identify the swine cardiovascular system based on input blood pressures to the heart. The blind identification method showed promising insight into the workings of the cardiovascular system based on a 7,000 sample test case.

The Minimum Model Error (MME) state estimation algorithm was used in [5] for estimating the states of poorly-model dynamic systems. The MME model initially estimates a model from a series of discrete measurements from the system,

then determines the un-modeled system dynamics based on parameter optimization to minimize a given cost function. The optimized parameters were then used to correct the initial model.

SVR, an application of Support Vector Machines (SVM) for function approximation was used in [49, 120]. SVR is preferable due to its high generalization capability [120].

Works by [70] applied Genetic Programming (GP) together with OLS algorithm to estimate compressive strength of carbon-fibre reinforced plastic. The authors found that the GP method was better than ANN for modeling uncertainties in the system, at the expense of computational cost and model non-transparency.

E. Structure Selection

Structure selection is defined as the task of selecting a subset of regressors to represent system dynamics [50, 97, 104, 105]. The objective of structure selection is model parsimony – the model should be able to explain the dynamics of the data using the least number of regressor terms [98, 105]. In its simplest form, the structure selection task involves the determination of the optimal lag space [51]. Selecting a higher lag space incorporates more history into the model leading to better prediction [105]. However, this approach increases the computational complexity of the model, as well as reducing the generalization ability of the model [1, 92, 104, 105]. Therefore, advanced methods utilize quality measures or criterions to select the most important regressor [50]. Several structure selection methods are deliberated in this section:

1. Trial and Error (Section 1)).
2. OLS (Section 2)).
3. Clustering (Section 3)).
4. Correlation-based (Section 4)).

Apart from the structure selection methods listed, other structure selection methods are described in Section 5).

1) Trial and Error Methods

Two trial and error methods for structure selection were found in literature. The first method called Zero-and-Refitting [50] estimates the parameters for a large model and gradually eliminate regressors that have sufficiently small parameter values. This method generally does not work well in practice as the parameter estimates are sensitive to the sampling window, noise variance and non-normalized data [50].

Another method was found in [61] for structure selection of a Bayesian NARX ANN. The method constructs multiple models with different input lags and hidden units in an effort to find the best ANN and data structure. This method works well if the number of parameter combinations is small, but can be overwhelming when the number of adjustable parameters grows. Similarly, in [85], an iterative two-stage NARX model construction method was presented, which created models by adding and deleting regressors to minimize model simulation error. The authors reported good accuracy with low computational cost.

2) OLS

The OLS algorithm is a simultaneous structure selection / parameter estimation algorithm introduced by [85, 133]. The algorithm works by first transforming the original regressors into orthogonal vectors. This step decouples the regressors (using methods such as Gram-Schmidt, Modified Gram-Schmidt, Householder transform, or Givens Rotation [49, 92, 100]) so that their individual contributions are estimated separately from the rest of the regressors [36, 85, 86, 94, 95, 97, 104].

The next step is to use the ERR measure to rank the contribution of each regressor towards reducing the approximation error of the model [85, 95, 104]. The ERR performs this by iteratively adding regressors from an initial empty model [94], and evaluating regressors combinations that has the most influence over the variance of the system output [70, 95]. Regressors with highest ERR values are deemed most significant and selected to be included in the final model structure [92]. The cutoff of selected regressors are determined based on a threshold [98].

Once the model structure has been determined, the parameters are estimated using Least-Squares (LS) or its variants [28, 82, 108, 134]. The OLS algorithm has since been widely accepted as a standard [6, 98, 104] and has been used in many works (see [28, 98, 108, 134] for examples, and [63, 122-124] for variations of applications) due to its simplicity, accuracy and effectiveness [1, 63].

Among the advantages of OLS are that the decoupled and decomposed nature of the orthogonalized regressors allow easy measurement and ranking of each regressor contributions [104]. Furthermore, OLS can structure selection without a priori knowledge of the system [27], a condition common in many real-life modeling scenarios. The algorithm has a successful track record in the field of SI [50].

Despite OLS's effectiveness, several criticisms have been directed towards its tendency to select excessive or sub-optimal terms [85, 92, 98, 104, 108], sensitivity to initialization and the order of regressor inclusion [85], repetitive orthogonalization procedure [49], and bias in regressor selection [97, 104]. Furthermore, a demonstration by [104] proves that OLS selected incorrect terms when the data is contaminated by certain noise sequences, or when the system input is poorly designed [104]. The predetermined threshold for selection of regressors also needs to be empirically or heuristically determined [98], although it is usually set to a sufficiently high value. Several authors have made attempts to address these issues (addition of generalized cross-validation with hypopaper testing [104], OLS with exhaustive searching [86], regularized OLS [123], incorporation of information criterions, correlation analysis and pruning methods [6, 85, 92, 97, 104], and hybrid OLS algorithms [70, 97]).

3) Clustering Methods

Clustering methods group data based on their similarities and performs structure selection by adding a cluster to or removing them from the model structure. Clustering methods are determined by two parameters, namely the number of clusters and the initial location of cluster centers [68].

In [50], nearest neighbor clustering was performed to determine the input/output lag space of a NARX model. After clustering, excess clusters can then be deleted from the model to achieve the desired model behavior. In [120], clustering was used to implement the optimal structure of a recurrent fuzzy ANN. The authors reported a reduction of the ANN size with the cluster-based structure determination method. A Gustafsson-Kessel fuzzy clustering algorithm was used in [68] to perform unsupervised data clustering, which were then modeled using local linear models. Clustering using Independent Component Analysis (ICA) has also been reported [30].

As reported by [50, 120], clustering improves the overall model structure either by removing excess terms that are collectively similar or grouping them together to create a compact classifier. However, a small number of references reviewed used clustering, which suggests that this type of method is in its infancy and does not possess a significant track record in current SI methodology.

4) Correlation-Based Methods

Correlation-based methods have been used in [92, 98, 104, 135]. Correlation-based methods reveal causal relationships between regressors [95], which can then be used to determine important terms to include in the model.

Correlation-based analysis guided by OLS was used in [92] to select significant model inputs for a polynomial model. Correlation analysis was performed to evaluate inputs that make a large contribution towards the output. The candidate inputs are refined further through the use of multi-objective evolutionary optimization to select the optimal model structure.

Works by [104] used a combination of generalized cross-validation, OLS and hypopaper testing to perform model structure selection. An integration between the adaptive orthogonal search algorithm and the Adjustable Prediction Error Sum of Squares (APRESS) was presented in [98] for structure selection. The authors reported that the simple algorithm achieved a good balance between reducing bias and improving variance of the model. Additionally, in [135], a gridding method combined with cross-correlation testing was used to determine the lags of an online SI model.

Reference [95] cautioned against the use of correlation-based methods because although correlations indicate causal relationship, qualitative analysis based on these methods are inaccurate for nonlinear systems modeling. The authors preferred the ERR analysis instead for their analysis.

5) Other Structure Selection Methods

This section reviews structure selection methods other than ones described in Section 1) and 3). A structure selection technique for time-series models was presented in [136]. The method used Monte-Carlo simulation and Analysis of Variance (ANOVA) sensitivity analysis to decompose the variance of regressors. The variance decompositions quantify the effect of variables in the model, and are used as a guide for structure selection.

F. Parameter Estimation

After the model structure has been determined, the next task is to estimate the parameters of the model. Among the methods surveyed, the most common was LS, followed by gradient-based algorithms, and others. These methods are described in Section 1). Several parameter estimation improvement methods are described in Section 2).

1) Parameter Estimation Techniques

The solution of LS problem is by far the most common parameter estimation method. The method involves adjusting the parameters of an over-determined system such that the sum of squares of the residual is minimized. LS methods are typically used in polynomial SI model as a form of curve-fitting so that the model fits the data from the original system.

The solution for LS problems is usually determined using three methods: normal equations, QR factorization and Singular Value Decomposition (SVD) [68, 137, 138]. Of these three, SVD is widely accepted to produce the most accurate results [139]. Several examples of LS for solving polynomial structures are given [25, 70, 85, 94, 97, 110]. Apart from that, other variations found were Generalized Total LS [31], Bias-Compensated LS [76], Extended LS [22, 79], Recursive LS [2, 49, 132, 140], and Nelder-Mead Simplex method [67]. Among the advantages of LS are its numerical stability and computational simplicity [141]. However, the results may be biased by noise [31, 142], or susceptible to ill-conditioning, leading to several methods and variants to specifically address the issue [31, 76, 139].

Gradient-based algorithms find optimal parameters by utilizing gradient information to traverse across and find minimas in the error surface. ANN-based methods are well-known for using gradient-based algorithms (see [28, 61, 113] (LM) and [93] (Gauss-Newton (GN)) for examples). Other variations include Mutual Information Gradient-Normalized Mean Squares [82] and Generalized Extended Stochastic Gradient [143].

Other methods found in literature presented a myriad of parameter estimation techniques: Time-Varying Minimization of Hyper-Surface Distance [142], Finite Element estimation [17], Canonical Parameterized Model [78], Extended Kalman Filter [81], ICA [30], Maximum Likelihood Estimate [135], MaxMI [82], and M-Law Memorized Improved Proportionate Affine Projection Algorithm (MMIPAPA) [144].

2) Improving Parameter Estimation

Techniques for improving parameters estimation are described in this section. The most common method reviewed was construction of piece-wise models, followed by pruning and regularization methods, and integration of known system behavior.

Piece-wise methods define individual linear sub-models for different system operating ranges [31, 46, 145]. This method has two primary benefits: improved model accuracy and transparency, noise resistance, reduced complexity, and solid foundation in linear modeling theory [7, 49, 68]. However, this method may be sub-optimal in systems that have strong nonlinearities or many operating regions [5]. Another

disadvantage is difficulty in transitioning between the sub-models [29, 46], which may lead to control instability [29]. In [86], variance minimization-based piecewise structure selection method was criticized for creating sub-optimal models. See [5, 7, 23, 27, 29, 46, 47, 49, 57, 68, 94, 145] for works that uses this method.

Another method involves integrating system knowledge into the parameter estimation process [31]. The model designers add useful system characteristics that can help model difficult dynamic processes. However, this method is inapplicable in systems where its dynamics is not fully understood.

Regularization and pruning methods have been used in [49, 69, 144]. In [144], a regularization parameter was added to the MMIPAPA parameter estimation algorithm. The authors reported that the use of regularization parameters allows a trade-off between rapid convergence and low estimation error. Pruning methods reduces the computational complexity and generalization of models by removing less important parameters [49]. Works by [49] utilized the moving window, while [69] used the Optimal Brain Damage (OBS) algorithm to remove superfluous weights from MLP models.

G. Model Validation

After the model parameters have been estimated, the model undergoes a validation process to ensure its representativeness of the actual system. Apart from curve-fitting testing methods normally used (such as Sum Squared Error (SSE), MSE, R-Squared and One-Step-Ahead (OSA) prediction, testing methods such as correlation tests also need to be applied. This is because a prerequisite for acceptance of the model should be the whiteness of the residuals [26, 98]. Additionally, information criterions can be used to quantify the model's complexity relative to its dataset fit. The information criterions are useful as models tend to over-fit the residuals if fitting accuracy is over-pursued [1].

OSA is used to measure the ability of the model to predict the one-step-ahead output based on past data [146]. Prediction over an extended time horizon is also available by applying OSA recursively, but they tend to diminish accuracy especially for highly nonlinear systems [116]. The OSA validation technique is described in Section 1).

The difference between the actual data and OSA predictions are called residuals. The residuals and be quantified using methods such as MSE, R-squared analysis and SSE. MSE calculates the mean, while SSE calculates the sum of squared residual values [147]. R-squared analysis is also a popular goodness of fit measure of the actual data based on SSE [148]. These techniques are described in Section 2).

An acceptable model and prediction results are only obtained if the model is unbiased [86]. If the model structure and parameter estimation is adequate (statistically valid), then the residuals should not contain any predictable terms across all past inputs and outputs [1, 86]. Correlation tests has been widely applied for SI, see [1] for linear and [86, 98, 101] for nonlinear application examples. Basic tests for linear systems are residuals autocorrelation test and cross-correlation test between residuals and the inputs [86], while extensions exist for non-linear cases (see [86, 98, 101]). Additionally, histogram tests on the residuals can be used to check the whiteness of the residuals. Correlation and histogram tests are described in Section 3) and Section 4), respectively.

Model parsimony is also an important element to consider in selecting the best model. The principle of parsimony is that in comparing between multiple model structures, the model that explains the dynamics of the system with the least complexity is the superior model. Information criterions such as Akaike Information Criterion (AIC), Final Prediction Error (FPE) and Model Descriptor Length (MDL) are used to enforce parsimony by incorporating complexity penalties in addition to residual fit [55, 64, 86]. The model selection criterions are described in Section 5).

1) One-Step-Ahead Prediction

OSA is a test that measures the ability of a model to predict future values based on its previous data. It takes the form of [146]:

$$\hat{y}(t) = \hat{g}(z(t)) \tag{16}$$

where $\hat{g}$ is the estimated nonlinear model, and $z(t)$ are the regressors. Sample representation of $z(t)$ for the NARX, NARMA and NARMAX models are shown in Eq. (17) to Eq. (19), respectively:

$$\begin{aligned} z(t) \quad = \quad & [y(t-1),\ y(t-2), \dots, y(t-n_y), \\ & u(t-n_k-1),\ u(t-n_k-2), \dots, u(t-n_k-n_u)] \end{aligned} \tag{17}$$

$$\begin{aligned} z(t) \quad = \quad & [y(t-1),\ y(t-2), \dots, y(t-n_y), \\ & \varepsilon(t-1),\ \varepsilon(t-2), \dots, \varepsilon(t-n_\varepsilon)] \end{aligned} \tag{18}$$

$$\begin{aligned} z(t) \quad = \quad & [y(t-1),\ y(t-2), \dots, y(t-n_y), \\ & u(t-n_k-1),\ u(t-n_k-2), \dots, u(t-n_k-n_u) \\ & \varepsilon(t-1),\ \varepsilon(t-2), \dots, \varepsilon(t-n_\varepsilon)] \end{aligned} \tag{19}$$

,

If necessary, a n-step ahead prediction may be obtained by iterative application of Eq. (17) to Eq. (19), substituting future outputs by OSA predictions [146].

2) Sum Squared Error, Mean-Squared Error and R-Squared

SSE and MSE are standard methods for testing the magnitude of residuals for regression and model fitting problems. The SSE equation for a residual vector ε of length n is given by [147]:

$$\begin{aligned} SSE \quad = \quad & \sum_{i=1}^{n} (\varepsilon_i)^2 \\ = \quad & \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \end{aligned} \tag{20}$$

where y_i is the observed value, and $\hat{y}_i$ is the estimated value at point i.

The MSE equation is similar to SSE, except the residuals are divided by n:

$$MSE = \frac{\sum_{i=1}^{n}(\varepsilon_i)^2}{n} = \frac{\sum_{i=1}^{n}(y_i-\hat{y}_i)^2}{n} \quad (21)$$

As the SSE and MSE values are calculated from the magnitude of residuals, low values indicate a good model fit. The ideal case for MSE and SSE is zero (when $y_i - \hat{y}_i = 0, i = 1, 2, \dots, n$). However, this rarely happens in actual modeling scenarios and a sufficiently small value is acceptable.

The R-squared technique is a measure to indicate the goodness-of-fit of a model. Originally intended for linear regression problems, recent works [52, 149] have extended its usage to measure nonlinear model fit as well. The R-squared measure is calculated as [149]:

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(y_i-\hat{y}_i)}{\sum_{i=1}^{N}(y_i-\bar{y})} \quad (22)$$

where y_i and $\hat{y}_i$ are actual and estimated observations at interval i, respectively. N is the number of observations and $\bar{y}$ is the mean value of y.

3) Correlation tests

Correlation tests measure the correlation between two time-series sequences at different points in time. They are useful indicators of dependencies and correlatedness between two sequences. Correlation tests are done by shifting the signals at different lags and measuring the correlation coefficients (degree of correlation) between them.

In SI, correlation tests are used to validate the model by determining the whiteness of its residuals. A residual sequence exhibits white noise characteristics if tests Eq. (23) to Eq. (27) hold [69]:

$$\theta_{\varepsilon\varepsilon}(\tau) = E[\varepsilon(t-\tau)\varepsilon(t)] = \delta(\tau) \quad (23)$$

$$\theta_{\varepsilon^2\varepsilon^2}(\tau) = E[\varepsilon^2(t-\tau)\varepsilon^2(t)] = \delta(\tau) \quad (24)$$

$$\theta_{y\varepsilon}(\tau) = E[y(t-\tau)\varepsilon(t)] = 0, \forall\tau \quad (25)$$

$$\theta_{y^2\varepsilon}(\tau) = E[(y^2(t-\tau) - \bar{y}^2(\tau))\varepsilon(t)] = 0, \forall\tau \quad (26)$$

$$\theta_{y^2\varepsilon^2}(\tau) = E[(y^2(t-\tau) - \bar{y}^2(\tau))\varepsilon^2(t)] = 0, \forall\tau \quad (27)$$

where:

$\theta_{x_1x_2}(\tau)$ = correlation coefficient between signals x_1 and x_2.

$E[\blacksquare]$ = mathematical expectation of the correlation function.

$\varepsilon(t)$ = model residuals = $y(t) - \hat{y}(t)$.

τ = lag space.

$y(t)$ = observed output at time t.

$\delta(\tau)$ = Kronecker delta, defined as:

$$\delta(\tau) = \begin{cases} 1, & \tau = 0 \\ 0, & \tau \neq 0 \end{cases} \quad (28)$$

The confidence band reveals the significance of the correlation, and a significantly large correlation is indicated by one or more coefficients lying outside the confidence band [86]. In correlation tests, the 95% confidence band is required because there is a finite amount of data length available [86]. The model is usually accepted if the correlation coefficients lie within the 95% confidence limits, defined as $\pm 1.96/n$, with n is the number of data points in the sequence.

4) Histogram Analysis

A histogram is a graphical method to present a distribution summary of a univariate data set [150]. It is drawn by segmenting the data into equal-sized bins (classes), then plotting the frequencies of data appearing in each bin. The horizontal axis of the histogram plot shows the bins, while the vertical axis depicts the data point frequencies.

In SI, histogram analysis is used to view the distribution of the residuals. The histogram exhibits white noise as a symmetric bell-shaped distribution with most of the frequency counts grouped in the middle and tapering off at both tails (Figure 3) [150].

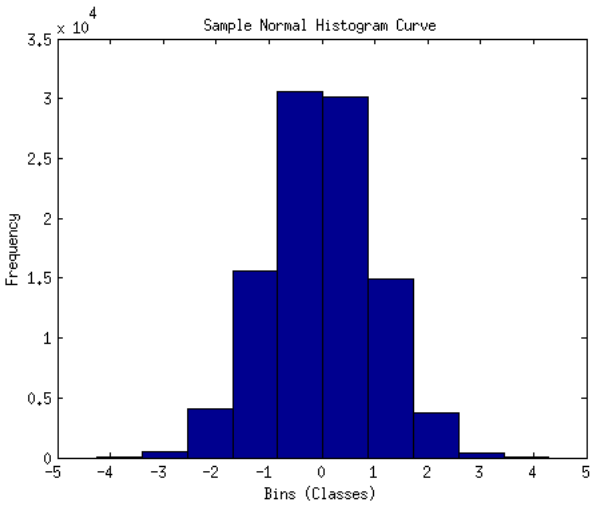

Figure 3: Sample of normally-distributed histogram curve for 10,000 random numbers.

5) Model Selection Criterions

The model selection criterions aim to assist practitioners in selecting a parsimonious model (a model that best represents the system with minimum number of parameters). The criterions do this by considering two aspects: the magnitude of the residuals, as well as the number of parameters selected. Each of the criterions are defined in Eq. (27) to Eq. (29) [55]:

$$V_{AIC} = \left(1 + 2\frac{d}{N}\right) V_{NSSE}(\theta, Z^N) \quad (29)$$

$$V_{MDL} = \left(1 + \log(N)\frac{d}{N}\right) V_{NSSE}(\theta, Z^N) \quad (30)$$

$$V_{FPE} = \left(\frac{1 + \frac{d}{N}}{1 - \frac{d}{N}}\right) V_{NSSE}(\theta, Z^N) \quad (31)$$

where d is the number of estimated parameters, N is the number of data points. $V_{NSSE}(\theta, Z^N)$ is the Normalized Sum Squared Error (NSSE) of the model residuals with respect to the model parameters, θ:

$$V_{NSSE}(\theta, Z^N) = \frac{1}{2N}\sum_{t=1}^{N} \varepsilon^2(t, \theta) \quad (32)$$

IV. Optimization and Its Application to SI

SI can be viewed as an optimization problem, where various models are evaluated based on objective functions forming a trade-off between model representativeness and various structural and parametric constraints [1].

Optimization is defined as a task to search for an optimal set of solutions for a parameter-dependent problem, typically by minimizing a cost function related to the problem [151]. Optimization techniques are divided into two categories, namely local and global techniques [152]. Local optimization techniques search the solution space based on gradient information. This technique benefits from rapid convergence. However, it only works well for a small number of parameters, as well as being limited by its initial conditions. Furthermore, the results are greatly affected by the presence of discontinuities in the data [152]. Examples of this type of optimization technique are GN and LM algorithms.

Global optimization techniques are more adept at handling multidimensional data and able to solve problems independent of initial values and under relaxed constraints and assumptions [152]. They achieve this through stochastic optimization process, where the algorithms emphasize exploration of the solution space to search for the best results [153]. Recent research has suggested that this type of optimization is superior in difficult problems [54], albeit slower [124]. Several notable algorithms in this category are Bacteria Foraging Algorithm (BFA) [154], Artificial Bee Colony (ABC) [155, 156], PSO, GA and Ant Colony Optimization (ACO) [122, 153, 154, 156, 157].

Typically, stochastic algorithms perform optimization by manipulating many basic autonomous individual units called agents [153]. These agents are represented in many forms: swarms in PSO, populations in GA and colonies in ACO and ABC. These agents perform basic tasks individually and interact with each other using simple protocols [153]. However, when combined, they exhibit a synergy of self-coordination that are able to solve optimization-type problems [153].

The global stochastic optimization technique was selected due to the high-dimensional nature of the problem presented in this paper. Following sections describe two stochastic algorithms extensively used for SI in literature: PSO (Section A) and GA (Section B).

A. PSO

PSO is a stochastic optimization algorithm that simulates the behavioral dynamics of swarming animals. The algorithm defines particles as agents responsible for searching the solution space. The movement of particles is directed by the particles' best individual achievement, as well as the best swarm achievement [54, 158]. The iterative search process continues until the objective is met or the number of iterations is exhausted.

The vanilla PSO algorithm suffered from suboptimal convergence because the particles tend to take large steps towards the end of the optimization [159], when ideally they should take smaller steps near the optima. Several variants have resolved this problem by adding a constriction term that gradually decreases the step size towards the end of the search (see [37, 112, 121, 126, 160-169] for implementations). However, the introduction of constriction may lead to another problem - premature convergence [159, 170-172]. Several new variations have been researched to resolve this issue, usually in combination with other optimization algorithms to form new hybridized variants [159, 161, 168, 173, 174], improving the algorithm with heuristics [166], inclusion of additional optimization parameters [37, 54, 165, 168, 171], or multiple swarms architecture [158, 172].

PSO is appealing to researchers because of several factors:

1. Usage of simple mathematical operators makes it easy to implement [37, 54, 112, 127, 159, 164, 170, 172, 175].
2. It is computationally inexpensive and fast compared to other more complex optimization algorithms [37, 54, 112, 127, 159, 161, 166, 171-173].
3. It has a successful track record in solving complex optimization problems [54, 158, 165, 168, 175].
4. It is versatile [168]: can be easily adapted to solve continuous, binary or discrete problems. No requirement of gradient information [37], thus can be used for a wide range of problems otherwise non-optimizable by gradient-based algorithms.
5. It requires a minimum amount of parameter adjustments [158, 166].
6. It is robust in solving complex optimization problem [121, 161].
7. It can be implemented in a true parallel fashion [176].

PSO has enjoyed significant interest in the field of SI. Almost all applications were limited to parameter estimation tasks – either for training ANNs [26, 37, 121, 128], or for estimation of its optimal training parameters [54, 127].

PSO was used to perform training on a WNN in [128] and RBFNN in [26, 121]. In [127], the PSO algorithm was used to optimize the learning rates of RBFNN for identification, while in [54], PSO was used to tune the RBFNN nodes together with the orthogonal forward regression algorithm.

Apart from RBFNN, Gaussian PSO was also used to adjust the weights of a Dynamic MLP for control of systems with long time delays [37]. The Gaussian PSO introduces chaos terms into the PSO algorithm to balance between exploration and exploitation of the algorithm.

A non-ANN application of PSO was found in [160], where a polynomial-based model was estimated using PSO to model a linear phased array pattern design with null steering model.

B. GA

GA is a stochastic optimization algorithm inspired by the concept of evolution and natural selection [54, 126, 167, 177]. The concept of natural selection is based on the theory that strong individuals are more likely to survive and reproduce, while weaker individuals are less likely to survive [10]. The offsprings of strong individuals inherit good traits from its parents, while weak individuals are removed from the gene pool as they die [10]. This mechanism allows only the strongest offsprings to survive until future generations.

This cooperation/competition behavior amongst the individuals is similar to probabilistic searching used in GA [10, 54]. GA defines its agents as a population of individuals [10] that carry certain traits [10, 62]. These traits are binary in nature, but they can be encoded such that they represent real numbers [58, 167, 178]. Each individual is ranked according to a fitness function at each generation [10, 167]. New generations are created from an initial population to create offsprings [126, 177].

These offsprings inherit certain traits from their previous generation through elitist/selection, crossover and mutation [10, 58, 62, 167, 178, 179]. These mechanisms are required for the offspring population diversity [167]. The elitist/selection mechanism selects the best individuals into the next generation through various strategies [10, 58, 126, 167]. Crossover combines parts of pairs of individuals to produce new offspring, while mutation adds a degree of diversity to the population by randomly altering the offspring [58, 126, 167]. Both crossover and mutation are useful to break free from local optimas [167, 168]. Optimization is continued until certain stopping criteria are reached [10, 179], with the best-performing individual(s) are accepted as the solution [168].

Among the advantages of GA are:

1. It does not require derivation or gradient information of the problem, thus suitable in problems where mathematical derivation is difficult [54, 180].
2. The algorithm is adaptive and fault-tolerant [175].
3. It can be implemented in a true parallel fashion [167, 176].
4. It is a simple, structured, efficient and accurate optimization algorithm [167, 176, 178, 180].

However, GA has been criticized for its slow convergence, which is a performance bottleneck in practical applications [167, 168]. Another drawback is that GA is susceptible to its optimization parameters, often resulting in premature convergence [62].

GA has enjoyed relatively higher exposure to the SI community compared to PSO, as evidenced by relevant papers published in the area. Three main applications of GA were found in literature – parameter optimization, parameter estimation and structure selection.

GA is an appealing choice for optimization of model parameters. This may be due to some models having many adjustment parameters that it is impossible to adequately determine the best combination on a trial-and-error basis.

In [62], a modified GA (with extinction, advanced elitist and bit-inversion mutation strategies) was used to optimize the parameters a fuzzy-based NARX model. Similarly, [131] and [99] utilized GA to optimize ANFIS training parameters, while several others [58, 126] used GA to optimize the parameters of a RBFNN.

GA has also been used for estimation of parameters in several SI models. GA was used to perform an initial parameter search prior to a gradient-based GN approach for identification of a synchronous generator [54]. In [62], a fuzzy logic-based identification model utilized GA to generate IF-THEN rules to characterize the behavior of a two-axis pneumatic artificial muscle robot arm, while [59] used GA to estimate the parameters of an Adaptive Backspace Control Model. Similarly, [10] reported using GA to estimate the parameters of a recurrent ANFIS model of a rainfall-runoff prediction model.

Two applications of GA for structure selection of SI models were found in [92, 104]. Works by [92], utilized GA to perform structure selection for a polynomial NARX model. In [104], a combination of GA and OLS was used to perform structure selection and parameter estimation, respectively. Both authors agree that GA imposed a heavy computational burden for the task [92, 104], albeit a reduced one compared to an exhaustive search approach [104].

V. Conclusions

SI is a discipline in control engineering concerned with representing a dynamic system using models. There exists a structured methodology for achieving the modeling task, which has been described in this paper. Several model types can be chosen from, and they can be represented with various estimators. Validation of the model is a crucial part of the identification process. Apart from conventional fitting tests, care must be taken to ensure that the residuals are uncorrelated. Correlated residuals would render the model invalid, thus the results unacceptable.

Several new directions in SI research involve the integration of stochastic optimization algorithms into the conventional SI methodology. This is because these type of optimization algorithms has been shown to outperform numerical techniques often used in SI. The review suggests that PSO and GA were the most common algorithms to be used for SI, and detailed descriptions of each have been presented.

References

[1] X. Hong, R. J. Mitchell, S. Chen, C. J. Harris, K.Li, and G. W. Irwin, "Model selection approaches for non-linear system identification: a review," *Int. J. of Systems Science*, vol. 39(10), pp. 925-946, 2008.

[2] F. Ding, P. X. Liu, and G. Liu, "Multiinnovation Least-Squares Identification for System Modeling," *IEEE Trans. Systems, Man & Cybernetics - Part B:Cybernetics*, vol. 40(3), pp. 767-778, 2010.

[3] D. T. Westwick and E. J. Perreault, "Closed-Loop Identification: Application to the Estimation of Limb Impedance in a Compliant Environment," *IEEE Trans. Biomedical Engineering*, vol. 58(3), pp. 521-530, 2011.

[4] O. A. Dahunsi, J. O. Pedro, and O. T. Nyandoro, "System Identification and Neural Network Based PID Control of Servo-Hydraulic Vehicle Suspension System," *SAIEE Africa Research J.*, vol. 101(3), pp. 93-105, 2010.

[5] J. R. Kolodziej and D. J. Mook, "Model determination for nonlinear state-based system identification," *Nonlinear Dynamics*, vol. 2011, pp. 735-753, 2011.

[6] X. Hong, S. Chen, and C. J. Harris, "Construction of Radial Basis Function Networks with Diversified Topologies," in *System identification, environmental modelling, and control system design*, L. Wang and H. Garnier, Eds., ed London: Springer, 2011, pp. 251-270.

[7] Y. Zhao, D. T. Westwick, and R. E. Kearney, "Subspace Methods for Identification of Human Ankle Joint Stiffness," *IEEE Trans. Biomedical Engineering*, vol. 58(11), pp. 3039-3048, 2011.

[8] M. A. Gonzales-Olvera and Y. Tang, "Black-Box Identification of a Class of Nonlinear Systems by a Recurrent Neurofuzzy Network," *IEEE Trans Neural Networks*, vol. 21(4), pp. 672-679, 2010.

[9] J.-O. Hahn, D. B. McCombie, A. T. Reisner, H. M. Hojman, and H. H. Asada, "Identification of Multichannel Cardiovascular Dynamics Using Dual Laguerre Basis Functions for Noninvasive Cardiovascular Monitoring," *IEEE Trans.Automatic Control Technology*, vol. 18(1), pp. 170-176, 2010.

[10] A. G. Evsukoff, B. S. L. P. d. Lima, and N. F. F. Ebecken, "Long-Term Runoff Modeling Using Rainfall Forecasts with Application to the Iguaçu River Basin," *Water Resource Management:*, vol. 25, pp. 963-985, 2011.

[11] M. Matus, D. Sáez, M. Favley, C. Suazo-Martínez, J. Moya, G. Jiménez-Estévez, R. Palma-Behnke, G. Olguín, and P. Jorquera, "Identification of Critical Spans for Monitoring Systems in Dynamic Thermal Rating," *IEEE Trans. Power Delivery*, vol. 27(2), pp. 1002-1009, 2012.

[12] J. Fei and M. Xin, "An adaptive fuzzy sliding mode controller for MEMS triaxial gyroscope with angular velocity estimation," *Nonlinear Dynamics*, pp. 1-13, 2012.

[13] S.-R. Oh, J. Sun, Z. Li, E. A. Celkis, and D. Parsons, "System Identification of a Model Ship Using a Mechatronic System," *IEEE/ASME Trans. Mechatronics*, vol. 15(2), pp. 316-320, 2010.

[14] T.-K. Dao and C.-K. Chen, "Path Tracking Control of a Motorcycle Based on System Identification," *IEEE Trans. Vehicular Technology*, vol. 60(7), pp. 2927-2935, 2011.

[15] S. O. D. Luiz, A. Perkusich, A. M. N. Lima, J. J. Silva, and H. Almeida, "System Identification and Energy-aware Processor Utilization Control," *IEEE Trans. Consumer Electronics*, vol. 58(1), pp. 32-37, 2012.

[16] P. V. Manivannan, M. Singaperumal, and A. Ramesh, "Development of an Idle Speed Engine Model using In-Cylinder Pressure Data and an Idle Speed Controller for a Small Capacity Port Fuel Injected SI Engine," *Int. J. of Automotive Technology*, vol. 12(1), pp. 11-20, 2011.

[17] D. Zhu, J. Guo, C. Cho, Y. Wang, and K.-M. Lee, "Wireless Mobile Sensor Network for the System Identification of a Space Frame Bridge," *IEEE/ASME Trans. on Mechatronics*, vol. 17(3), pp. 499-507, 2012.

[18] L. YuMin, T. DaZhen, X. Hao, and T. Shu, "Productivity matching and quantitative prediction of coalbed methane wells based on BP neural network," *Science China Technological Sciences*, vol. 54(5), pp. 1281-1286, 2011.

[19] J. W. Pierre, N. Zhou, F. K. Tuffner, J. F. Hauer, D. J. Trudnowski, and W. A. Mittelstadt, "Probing Signal Design for Power System Identification," *IEEE Trans. Power Systems*, vol. 25(2), pp. 835-843, 2010.

[20] S. D. Keen and D. J. Cole, "Bias-Free Identification of a Linear Model-Predictive Steering Controller From Measured Driver Steering Behavior," *IEEE Trans. Systems, Man & Cybernetics - Part B:Cybernetics*, vol. 42(2), pp. 434-443, 2012.

[21] E.-W. Bai, "Non-Parametric Nonlinear System Identification: An Asymptotic Minimum Mean Squared Error Estimator," *IEEE Trans. Automatic Control*, vol. 55(7), pp. 1615-1626, 2010.

[22] A. Ghoreyshi and H. L. Galiana, "Simultaneous Identification of Oculomotor Subsystems Using a Hybrid System Approach: Introducing Hybrid Extended Least Squares," *IEEE Trans. Biomedical Engineering*, vol. 57(5), pp. 1089-1098, 2010.

[23] Y. S. Lee, S. Tsakirtzis, A. F. Vakakis, L. A. Bergman, and D. M. McFarland, "A time-domain nonlinear system identification method based on multiscale dynamic partitions," *Meccanica*, vol. 46, pp. 625-649, 2011.

[24] W.-X. Zhao, H.-F. Chen, and W. X. Zheng, "Recursive Identification for Nonlinear ARX Systems Based on Stochastic Approximation Algorithm," *IEEE Trans. Automatic Control*, vol. 55(6), pp. 1287-1299, 2010.

[25] M. M. Peretz and S. Ben-Yaakov, "Time-domain identification of pulse-width modulated converters," *IET Power Electronics*, vol. 5(2), pp. 166 -172, 2012.

[26] M. D. Spiridonakos, A. G. Poulimenos, and S. D. Fassois, "Output-Only Identification and Dynamic Analysis of Time-Varying Mechanical Structures Under Random Excitation: A comparative assessment of parametric methods," *J. Sound and Vibration*, vol. 329, pp. 768-785, 2010.

[27] E.-W. Bai and M. Deistler, "An Interactive Term Approach to Non-Parametric FIR Nonlinear System Identification," *IEEE Trans. Automatic Control*, vol. 55(8), pp. 1952-1957, 2010.

[28] B. H. G. Barbosa, L. A. Aguirre, C. B. Martinez, and A. P. Braga, "Black and Gray-Box Identification of a Hydraulic Pumping System," *IEEE Trans. Control Systems Technology*, vol. 19(2), pp. 398-406, 2011.

[29] S. S. Ahmadi and M. Karrari, "Nonlinear identification of synchronous generators using a local model approach," *Przeglad Elektrotechniczny (Electrical Review)*, vol. 8, pp. 166-170, 2011.

[30] R. Talmon, D. Kushnir, R. R. Coifman, I. Cohen, and S. Gannot, "Parametrization of Linear Systems Using Diffusion Kernels," *IEEE Trans. Signal Processing*, vol. 60(3), 2012.

[31] C. Hametner and S. Jakubek, "Nonlinear Identification with Local Model Networks Using GTLS Techniques and Equality Constraints," *IEEE Trans Neural Networks*, vol. 22(9), pp. 1406-1418, 2011.

[32] N. Dong and Z. Chen, "A novel data based control method based upon neural network and simultaneous perturbation stochastic approximation," *Nonlinear Dynamics*, vol. 67, pp. 957-963, 2012.

[33] T. Schwelckhardt and F. Allgower, "Good or Bad - When is Plant Nonlinearity an Obstacle for Control?," in *Int. Symp. on Advanced Control of Chemical Processes (ADCHEM 2006)*, Gramado, Brazil, 2006, pp. 37-44.

[34] A. Novak, L. Simon, F. Kadlec, and P. Lotton, "Nonlinear System Identification Using Exponential Swept-Sine Signal," *IEEE Trans. Instrumentation & Measurement*, vol. 59(8), pp. 2220-2229, 2010.

[35] Y. Avargel and I. Cohen, "Modeling and Identification of Nonlinear Systems in the Short-Time Fourier Transform Domain," *IEEE Trans. Signal Processing*, vol. 58(1), pp. 291-304, 2010.

[36] A. Y. Kibangou and G. Favier, "Identification of fifth-order Volterra systems using i.i.d. inputs," *IET Signal Processing*, vol. 4(1), pp. 30-44, 2010.

[37] M. Han, J. Fan, and J. Wang, "A Dynamic Feedforward Neural Network Based on Gaussian Particle Swarm Optimization and its Application for Predictive Control," *IEEE Trans. Neural Networks*, vol. 22(9), pp. 1457-1468, 2011.

[38] L. d. S. Coelho and R. A. Krohling, "Nonlinear system identification based on B-spline neural network and modified particle swarm optimization," presented at the Int. Joint Conf. on Neural Networks, Vancouver, Canada, 2006.

[39] N. Chiras, C. Evans, and D. Rees, "Nonlinear gas turbine modeling using NARMAX structures," *IEEE Trans. Instrumentation and Measurement*, vol. 50(4), pp. 893-898, 2001.

[40] Y. Bi, Y. Zhu, and Y. Jiang, "A new nonlinear system identification method using gene expression programming," presented at the Proc. of IEEE Int. Conf. on Mechatronics and Automation, Harbin, China, 2007.

[41] N. R. Butt, M. Shafiq, and T. Khan, "An adaptive root-solving controller for tracking of nonlinear dynamic plants," *Proc. Int. Conf. on Industrial Electronics and Applications*, pp. 6-6, 2005.

[42] M. Solomou and D. Rees, "System modeling in the presence of nonlinear distortions," presented at the Instrumentation and Measurement Technology Conf., Lake Como, Italy, 2004.

[43] C. A. Sierakowski, F. A. Guerra, and L. d. S. Coelho, "Particle swarm optimization approach for multi-step ahead prediction using radial basis function neural network," presented at the Proc. 15th IFAC World Congress, Praga, Portugal, 2005.

[44] L. K. Chen and A. G. Ulsoy, "Identification of a nonlinear driver model via NARMAX modeling," presented at the Proc. American Control Conf., Chicago, IL, 2000.

[45] H.-F. Chen, "New Approach to Recursive Identification for ARMAX Systems," *IEEE Trans. Automatic Control*, vol. 55(4), pp. 868-879, 2010.

[46] C. Y. Lai, C. Xiang, and T. H. Lee, "Data-Based Identification and Control of Nonlinear Systems via Piecewise Affine Approximation," *IEEE Trans Neural Networks*, vol. 22(12), pp. 2189-2200, 2011.

[47] V. L. Le, G. Bloch, and F. Lauer, "Reduced-Size Kernel Models for Nonlinear Hybrid System Identification," *IEEE Trans. Neural Networks*, vol. 22(12), pp. 2398-2405, 2011.

[48] X.-M. Chen and H.-F. Chen, "Recursive Identification for MIMO Hammerstein Systems," *IEEE Trans. Automatic Control*, vol. 56(4), pp. 895-902, 2011.

[49] H. Ning, X. Jing, and L. Cheng, "Online Identification of Nonlinear Spatiotemporal Systems Using Kernel Learning Approach," *IEEE Trans. Neural Networks*, vol. 22(9), pp. 1381-1394, 2011.

[50] L. A. Aguirre and C. Letellier, "Modeling Nonlinear Dynamics and Chaos: A Review," *Mathematical Problems in Engineering*, vol. 2009, pp. 1-35, 2009.

[51] T. Cassar, K. P. Camilleri, and S. G. Fabri, "Order Estimation of Multivariate ARMA Models," *IEEE J. Selected Topics in Signal Processing*, vol. 4(3), pp. 494-503, 2010.

[52] B. A. Amisigo, N. v. d. Giesen, C. Rogers, W. E. I. Andah, and J. Friesen, "Monthly streamflow prediction in the Volta Basin of West Africa: A SISO NARMAX polynomial modelling," *Physics and Chemistry of the Earth*, vol. 33(1-2), pp. 141-150, 2007.

[53] X. Zheng and B. M. Chen, "Modeling and Forecasting of Stock Markets under a System Adaptation Framework," *J. Syst. Sci. Complex*, vol. 25, pp. 641-674, 2011.

[54] M. A. Arjona, M. Cisneros-González, and C. Hernández, "Parameter Estimation of a Synchronous Generator Using a Sine Cardinal Perturbation and Mixed Stochastic–Deterministic Algorithms," *IEEE Trans. Industrial Electronics*, vol. 58(2), pp. 486-493, 2011.

[55] M. H. F. Rahiman, "System Identification of Essential Oil Extraction System," Ph. D. Dissertation, Faculty of Electrical Engineering, Universiti Teknologi MARA, Shah Alam, Malaysia, 2008.

[56] N. A. Rahim, "The design of a non-linear autoregressive moving average with exegenous input (NARMAX) for a DC motor," MSc, Faculty of Electrical Engineering, Universiti Teknologi MARA, Shah Alam, 2004.

[57] G. V. Kaigala, J. Jiang, C. J. Backhouse, and H. J. Marquez, "System Design and Modeling of a Time-Varying, Nonlinear Temperature Controller for Microfluidics," *IEEE Trans. Control Systems Technology*, vol. 18(2), pp. 521-530, 2010.

[58] X. Zong-yi, Q. Yong, P. Xue-miao, J. Li-min, and Z. Yuan, "Modelling of the Automatic Depth Control Electrohydraulic System Using RBF Neural Network and Genetic Algorithm," *Mathematical Problems in Engineering*, vol. 2010, pp. 1-16, 2010.

[59] R.-F. Fung and W.-C. Lin, "System Identification and Contour Tracking of a Plane-Type 3-DOF (X, Y, Θ_z) Precision Positioning Table," *IEEE Trans. Control Systems Technology*, vol. 18(1), pp. 22-34, 2010.

[60] T. Oomen, J. v. d. Wijdeven, and O. H. Bosgra, "System Identification and Low-Order Optimal Control of Intersample Behavior in ILC," *IEEE Trans. Automatic Control*, vol. 56(11), pp. 2734-2739, 2011.

[61] Z. H. Chen and Y. Q. Ni, "On-board Identification and Control Performance Verification of an MR Damper Incorporated with Structure," *J. Intelligent Material Systems and Structures*, vol. 22, pp. 1551-1565, 2011.

[62] K. K. Ahn and H. P. H. Anh, "Inverse Double NARX Fuzzy Modeling for System Identification," *IEEE/ASME Trans. on Mechatronics*, vol. 15(1), pp. 136-148, 2010.

[63] M. J. Er, F. Liu, and M. B. Li, "Self-constructing Fuzzy Neural Networks with Extended Kalman Filter," *Int. J. Fuzzy Systems*, vol. 12(1), pp. 66-72, 2010.

[64] S. Haykin, *Neural Networks: A Comprehensive Foundation*. Delhi, India: Pearson Education (Singapore) Pte Ltd., 1999.

[65] D. Wang, X.-J. Zeng, and J. A. Keane, "An Evolving-Construction Scheme for Fuzzy Systems," *IEEE Trans. Fuzzy Systems*, vol. 18(4), pp. 755-770, 2010.

[66] D. Spina, B. G. Lamonaca, M. Nicoletti, and M. Dolce, "Structural monitoring by the Italian Department of Civil Protection and the case of 2009 Abruzzo seismic sequence," *Bull. Earthquake Engineering*, vol. 9, 2011.

[67] S. R. Anderson, N. F. Lepora, J. Porrill, and P. Dean, "Nonlinear Dynamic Modeling of Isometric Force Production in Primate Eye Muscle," *IEEE Trans. Biomedical Engineering*, vol. 57(7), pp. 1554-1567, 2010.

[68] L. Tesli, B. Hartmann, O. Nelles, and I. Škrjanc, "Nonlinear System Identification by Gustafson–Kessel Fuzzy Clustering and Supervised Local Model Network Learning for the Drug Absorption Spectra Process," *IEEE Trans. Neural Networks*, vol. 22(12), pp. 1941-1951, 2011.

[69] M. Norgaard, O. Ravn, N. K. Poulsen, and L. K. Hansen, *Neural networks for modeling and control of dynamic systems: A practitioner's handbook*. London: Springer, 2000.

[70] A. H. Gandomi, A. H. Alavi, and P. Arjmandi, "Genetic Programming and Orthogonal Least Squares: A Hybrid Approach to Compressive Strength of CFRP-confined Concrete Cylinders Modeling," *J. Mechanics of Materials & Structures*, vol. 5(5), pp. 735–753, 2010.

[71] B. Cosenza, "Development of a neural network for glucose concentration prevision in patients affected by type 1 diabetes," *Bioprocess & Biosystems Engineering*, pp. 1-9, 2012.

[72] M. H. Beale, M. T. Hagan, and H. B. Demuth, *Neural Network Toolbox User's Guide*. Natick, MA: Mathworks Inc., 2011.

[73] G. Kerschen, K. Worden, A. F. Vakakis, and J.-C. Golinval, "Past, present and future of nonlinear system identification in structural dynamics," *Mechanical Systems & Signal Processing*, vol. 20, pp. 505-592, 2006.

[74] A.-R. Amini, "Analysis of Power Transistor Behavioural Modeling Techniques Suitable for Narrow-band Power Amplifier Design " MSc. Thesis, University of Waterloo, Ontario, Canada, 2012.

[75] Y. Avargel and I. Cohen, "Undermodeling-Error Quantification for Quadratically Nonlinear System Identification in the Short-Time Fourier Transform Domain," *IEEE Trans. Signal Processing*, vol. 58(12), pp. 6052-6065, 2010.

[76] J. H. Kim and S. W. Nam, "Bias-compensated identification of quadratic Volterra system with noisy input and output," *Electronics Letters*, vol. 46(6), 2010.

[77] X. Ren and X. Lv, "Identification of Extended Hammerstein Systems Using Dynamic Self-Optimizing Neural Networks," *IEEE Trans Neural Networks*, vol. 22(8), pp. 1169-1179, 2011.

[78] X. Lv and X. Ren, "Non-iterative identification and model following control of Hammerstein systems with asymmetric dead-zone non-linearities," *IET Control Theory & Applications*, vol. 6(1), pp. 84– 89, 2012.

[79] W.-X. Zhao, "Parametric Identification of Hammerstein Systems With Consistency Results Using Stochastic Inputs," *IEEE Trans. Automatic Control*, vol. 55(2), pp. 474-480, 2010.

[80] E. Abd-Elrady and L. Gan, "Identification of Hammerstein and Wiener Models Using Spectral Magnitude Matching," *Proc. 17th IFAC World Congress*, vol. 17(1), 2008.

[81] M. M. d. Silva, T. Wigren, and T. Mendonça, "Nonlinear Identification of a Minimal Neuromuscular Blockade Model in Anesthesia," *IEEE Trans. Control Systems Technology*, vol. 20(1), pp. 181-188, 2012.

[82] B. Chen, Y. Zhu, J. Hu, and J. C. Prıncipe, "Stochastic gradient identification of Wiener system with maximum mutual information criterion," *IET Signal Processsing*, vol. 5(6), pp. 589- 597, 2011.

[83] F. Guo, "A New Identification Method for Wiener and Hammerstein Systems," Ph.D Dissertation, Institut fur Angewandte Informatik, University of Karlsruhe, Karlsruhe, Germany, 2004.

[84] I. J. Leontaritis and S. A. Billings, "Input-output parametric models for nonlinear systems," *Int. J. Contr.*, vol. 41, no. 2, pp. 311-341, 1985.

[85] L. Piroddi, "Simulation error minimisation methods for NARX model identification," *Int. J. Modelling, Identification and Control*, vol. 3(4), pp. 392-403, 2008.

[86] E. M. A. M. Mendes and S. A. Billings, "An Alternative Solution to the Model Structure Selection Problem," *IEEE Trans. Systems, Man & Cybernetics - Part A: Systems and Humans*, vol. 31(6), pp. 597-608, 2001.

[87] E. M. A. M. Mendes and S. A. Billings, "An alternative solution to the model structure selection problem," *IEEE Trans. Systems, Man and Cybernetics - Part A: Systems and Humans*, vol. 31(6), pp. 597-608, 2001.

[88] M. Vallverdu, M. J. Korenberg, and P. Caminal, "Model identification of the neural control of the cardiovascular system using NARMAX models," presented at the Proc. Computer in Cardiology, 1991.

[89] S. L. Kukreja, H. L. Galiana, and R. E. Kearney, "NARMAX representation and identification of ankle dynamics," *IEEE Trans. Biomedical Eng.*, vol. 50(1), pp. 70-81, 2003.

[90] S. L. Kukreja, R. E. Kearney, and H. L. Galiana, "A least squares parameter estimation algorithm for switched hammerstein systems with applications to the VOR," *IEEE Trans. Biomedical Eng.*, vol. 52(3), pp. 431-444, 2005.

[91] N. A. Rahim, M. N. Taib, and M. I. Yusof, "Nonlinear system identification for a DC motor using NARMAX Approach," in *Asian Conference on Sensors (AsiaSense)*, 2003, pp. 305 - 311

[92] Y. Cheng, L. Wang, M. Yu, and J. Hu, "An efficient identification scheme for a nonlinear polynomial NARX model," *Artificial Life Robotics*, vol. 16, pp. 70-73, 2011.

[93] M. Shafiq and N. R. Butt, "Utilizing Higher-Order Neural Networks in U-model Based Controllers for Stable Nonlinear Plants," *Int. J. Control, Automation & Systems*, vol. 9(3), pp. 489-496, 2011.

[94] L. Piroddi and M. Lovera, "NARX model identification with error filtering," in *Proc. 17th World Congress Int. Federation of Automatic Control*, Seoul, Korea, 2008, pp. 2726-2731.

[95] M. A. Balikhin, R. J. Boynton, S. N. Walker, J. E. Borovsky, S. A. Billings, and H. L. Wei, "Using the NARMAX approach to model the evolution of energetic electrons fluxes at geostationary orbit," *Geophysical Research Letters*, vol. 38(L18105), pp. 1-5, 2011.

[96] S. Chatterjee, S. Nigam, J. B. Singh, and L. N. Upadhyaya, "Software fault prediction using Nonlinear Autoregressive with eXogenous Inputs (NARX) network," *Applied Intelligence*, pp. 1-9, 2011.

[97] L. Piroddi, "Adaptive model selection for polynomial NARX models " *IET Control Theory & Applications*, vol. 4(12), pp. 2693-2706, 2010.

[98] H. L. Wei and S. A. Billings, "An adaptive orthogonal search algorithm for model subset selection and non-linear system identification," *Int. J. Control*, vol. 81(5), pp. 714-724, 2008.

[99] C. Safak, V. Topuz, and A. F. Baba, "Pneumatic motor speed control by trajectory tracking fuzzy logic controller," *Sadhana*, vol. 35(1), pp. 75-86, 2010.

[100] S. Chen, S. A. Billings, and W. Luo, "Orthogonal least squares methods and their application to non-linear system identification," *Int. J. Control*, vol. 50(5), pp. 1873 – 1896, 1989.

[101] S. A. Billings and S. Chen, "Extended model set, global data and threshold model identification of severely non-linear systems," *Int. J. of Control*, vol. 50(5), pp. 1897-1923, 1989.

[102] H. Wang, Y. Shen, T. Huang, and Z. Zeng, "Nonlinear System Identification based on Recurrent Wavelet Neural Network," in *Advances in Soft Computing*. vol. 56, ed: Springer, 2009.

[103] S. A. Billings and H.-L. Wei, "A New Class of Wavelet Networks for Nonlinear System Identification," *IEEE Trans. Neural Networks*, vol. 16(4), pp. 862-874, 2005.

[104] H. L. Wei and S. A. Billings, "Model structure selection using an integrated forward orthogonal search algorithm assisted by squared correlation and mutual information," *Int. J. Modeling, Identification & Control*, vol. 3(4), pp. 341-356, 2008.

[105] R. Amrit and P. Saha, "Stochastic identification of bioreactor process exhibiting input multiplicity," *Bioprocess & Biosystems Engineering*, vol. 30, pp. 165-172, 2007.

[106] L.-H. Geng, D.-Y. Xiao, T. Zhang, and J.-Y. Song, "Worst-Case Identification of Errors-in-Variables Models in

Closed Loop," *IEEE Trans. Automatic Control*, vol. 56(4), pp. 762-771, 2011.

[107] G. Palli, G. Borghesan, and C. Melchiorri, "Modeling, Identification, and Control of Tendon-Based Actuation Systems," *IEEE Trans. Robotics*, vol. 28(2), pp. 277-290, 2012.

[108] L. A. Aguirre and E. C. Furtado, "Building dynamical models from data and prior knowledge: The case of the first period-doubling bifurcation," *Physical Review*, vol. 76, pp. 046219-1 - 046219-13, 2007.

[109] J. D. Gabano, T. Poinot, and H. Kanoun, "Identification of a thermal system using continuous linear parameter-varying fractional modelling," *IET Control Theory Applications*, vol. 5(7), pp. 889–899, 2011.

[110] M. T. Rashid, M. Frasca, A. A. Ali, R. S. Ali, L. Fortuna, and M. G. Xibilia, "Nonlinear model identification for Artemia population motion," *Nonlinear Dynamics*, pp. 1-7, 2012.

[111] A. H. Tan and T. T. V. Yap, "Signal design for the identification of multivariable ill-conditioned systems using virtual transfer function between inputs," *IET Control Theory Applications*, vol. 6(3), pp. 394-402, 2012.

[112] T. Y.-B. a. X. Zhi-Bin, "Particle-swarm-optimization-based selective neural network ensemble and its application to modeling resonant frequency of microstrip antenna," *Microstrip Antennas*, pp. 69-82, 2011.

[113] N. Boëly and R. M. Botez, "New Approach for the Identification and Validation of a Nonlinear F/A-18 Model by Use of Neural Networks," *IEEE Trans Neural Networks*, vol. 21(11), pp. 1759-1765, 2010.

[114] J. Michalkiewicz, "Modified Kolmogorov's Neural Network in the Identification of Hammerstein and Wiener Systems," *IEEE Trans. Neural Networks & Learning Systems*, vol. 23(4), pp. 657-662, 2012.

[115] G.-X. Wen and Y.-J. Liu, "Adaptive fuzzy-neural tracking control for uncertain nonlinear discrete-time systems in the NARMAX form," *Nonlinear Dynamics*, vol. 66, pp. 745-753, 2011.

[116] P. Gómez-Gil, J. M. Ramírez-Cortes, S. E. P. Hernández, and V. Alarcón-Aquino, "A Neural Network Scheme for Long-Term Forecasting of Chaotic Time Series," *Neural Processing Letters*, vol. 33, pp. 215-233, 2011.

[117] K. Tanabe, B. Lucic, D. Ami, T. Kurita, M. Kaihara, N. Onodera, and T. Suzuki, "Prediction of carcinogenicity for diverse chemicals based on substructure grouping and SVM modeling," *Molecular Diversity*, vol. 14, pp. 789-802, 2010.

[118] R. Singh, E. Manitsas, B. C. Pal, and G. Strbac, "A Recursive Bayesian Approach for Identification of Network Configuration Changes in Distribution System State Estimation," *IEEE Trans. Power Systems*, vol. 25(3), pp. 1329-1336, 2010.

[119] M. Norgaard, O. Ravn, N. K. Poulsen, and L. K. Hansen, *Neural Networks for modelling and control of dynamic systems*. London: Springer-Verlag London Ltd., 2000.

[120] C.-F. Juang and C.-D. Hsieh, "A Locally Recurrent Fuzzy Neural Network With Support Vector Regression for Dynamic-System Modeling," *IEEE Trans. Fuzzy Systems*, vol. 18(2), pp. 261-273, 2010.

[121] M. Sheikhan, R. Shahnazi, and S. Garoucy, "Hyperchaos synchronization using PSO-optimized RBF-based controllers to improve security of communication systems," *Neural Computing & Applications*, pp. 1-12.

[122] W. Yunna and S. Zhaomin, "Application of Radial Basis Function Neural Network Based on Ant Colony Algorithm in Credit Evaluation of Real Estate Enterprises," in *Proc. 2008 Int. Conf. on Management Science and Engineering*, 2008, pp. 1322-1329.

[123] Y. W. Wong, K. P. Seng, and L.-M. Ang, "Radial Basis Function Neural Network With Incremental Learning for Face Recognition," *IEEE Trans. Systems, Man & Cybernetics - Part B:Cybernetics*, vol. 41(4), pp. 940-949, 2011.

[124] C.-C. Liao, "Enhanced RBF Network for Recognizing Noise-Riding Power Quality Events," *IEEE Trans. Instrumentation & Measurement*, vol. 59(6), pp. 1550-1561, 2010.

[125] S. Chen, X. Hong, and C. J. Harris, "Particle Swarm Optimization Aided Orthogonal Forward Regression for Unified Data Modeling," *IEEE Trans. Evolutionary Computation*, vol. 14(4), pp. 477-499, 2010.

[126] M. Sheikhan, R. Shahnazi, and S. Garoucy, "Synchronization of general chaotic systems using neural controllers with application to secure communication," *Neural Computing & Applications*, pp. 1-13, 2011.

[127] C.-N. Ko, "Identification of non-linear systems using radial basis function neural networks with time-varying learning algorithm," *IET Signal Processing*, vol. 6(2), pp. 91 –98, 2012.

[128] S. Yilmaz and Y. Oysal, "Fuzzy Wavelet Neural Network Models for Prediction and Identification of Dynamical Systems," *IEEE Trans Neural Networks*, vol. 21(10), pp. 1599-1609, 2010.

[129] M. Khezri and M. Jahed, "A Neuro–Fuzzy Inference System for sEMG-Based Identification of Hand Motion Commands," *IEEE Trans. Industrial Electronics*, vol. 58(5), pp. 1952-1960, 2011.

[130] M. Ayati, "Adaptive Fuzzy Control of Nonlinear in Parameters Uncertain Chaotic Systems Using Improved Speed Gradient Method," *Circuits System Signal Process*, vol. 31, pp. 911-926, 2012.

[131] C.-H. Ting, M. Mahfouf, A. Nassef, D. A. Linkens, George Panoutsos, P. Nickel, A. C. Roberts, and G. R. J. Hockey, "Real-Time Adaptive Automation System Based on Identification of Operator Functional State in Simulated Process Control Operations," *IEEE Trans. Systems, Man & Cybernetics - Part A: Systems and Humans*, vol. 40(2), pp. 251-262, 2010.

[132] A. Lemos, W. Caminhas, and F. Gomide, "Multivariable Gaussian Evolving Fuzzy Modeling System," *IEEE Trans. Fuzzy Systems*, vol. 19(1), pp. 91-104, 2011.

[133] S. A. Billings, M. Korenberg, and S. Chen, "Identification of nonlinear output-affine systems using an orthogonal least-squares algorithm," *J. Syst. Sci.*, vol. 19(8), pp. 1559–1568, 1988.

[134] D. G. Dimogianopoulos, J. D. Hios, and S. D. Fassois, "FDI for Aircraft Systems Using Stochastic Pooled-NARMAX Representations: Design and Assessment," *IEEE Trans. Control Systems Technology*, vol. 17(6), pp. 1385-1397, 2009.

[135] A. H. Tan and C. L. Cham, "Continuous-time model identification of a cooling system with variable delay," *IET Control Theory Applications*, vol. 5(7), pp. 913–922, 2011.

[136] T. J. Harris and W. Yu, "Variance decompositions of nonlinear time series using stochastic simulation and sensitivity analysis," *Statistical Computing*, vol. 22, pp. 387-396, 2012.

[137] M. Baboulin, S. Gratton, R. Lacroix, and A. J. Laub, "Efficient computation of condition estimates for linear least squares problems," Bâtiment Alan Turing, Campus de l'École Polytechnique, Palaiseau, France 2012.

[138] E. Fuchs, T. Gruber, J. Nitschke, and B. Sick, "Online Segmentation of Time Series Based on Polynomial Least-Squares Approximations," *IEEE Trans. Pattern Analysis & Machine Intelligence*, vol. 32(12), pp. 2232-2245, 2010.

[139] B. Nouri, R. Achar, and M. S. Nakhla, "z -Domain Orthonormal Basis Functions for Physical System

Identifications," *IEEE Trans. Advanced Packaging*, vol. 33(1), pp. 293-307, 2010.

[140] A. Chiuso, R. Muradore, and E. Marchetti, "Dynamic Calibration of Adaptive Optics Systems: A System Identification Approach," *IEEE Trans. Control Systems Technology*, vol. 18(3), pp. 705-713, 2010.

[141] Y. Zhou, S. C. Chan, and K. L. Ho, "New Sequential Partial-Update Least Mean M-Estimate Algorithms for Robust Adaptive System Identification in Impulsive Noise," *IEEE Trans. Industrial Electronics*, vol. 58(9), pp. 4455-4470, 2011.

[142] B. Yang and K. H. Chon, "Estimating Time-Varying Nonlinear Autoregressive Model Parameters by Minimizing Hypersurface Distance," *IEEE Trans. Biomedical Engineering*, vol. 57(8), pp. 1937-1944, 2010.

[143] L. Xie, Y. J. Liu, H. Z. Yang, and F. Ding, "Modelling and identification for non-uniformly periodically sampled-data systems," *IET Control Theory and Applications*, vol. 4(5), pp. 784-794, 2010.

[144] L. Xiao, Y. Wang, P. Zhang, M. Wu, and J. Yang, "Variable regularisation efficient m-law improved proportionate affine projection algorithm for sparse system identification," *Electronics Letters*, vol. 48(3), pp. 1-2, 2012.

[145] N. Ozay, M. Sznaier, C. M. Lagoa, and O. I. Camps, "A Sparsification Approach to Set Membership Identification of Switched Affine Systems," *IEEE Trans. Automatic Control*, vol. 57(3), pp. 634-648, 2012.

[146] M. Espinoza, J. A. K. Suykens, and B. D. Moor, "Kernel Based Partially Linear Models and Nonlinear Identification," *IEEE Trans. Automatic Control*, vol. 50(10), pp. 1602-1606, 2005.

[147] D. A. Konovalov, D. Coomans, E. Deconinck, and Y. V. Heyden, "Benchmarking of QSAR Models for Blood-Brain Barrier Permeation," *J. Chemical Information & Modeling*, vol. 47(4), pp. 1648-1656, 2007.

[148] C. A. Cameron and F. A. W. . "R-squared measures for count data regression models with applications to health-care utilization," *J. of Business & Economic Statistics*, vol. 14(2), pp. 209-220, 1996.

[149] M. H. F. Rahiman, M. N. Taib, and Y. M. Salleh, "Selection of training data for modeling essential oil extraction system using NNARX structure," in *2007 Int. Conf. Control, Automation & Systems (COEX 2007)*, Seoul, Korea, 2007, pp. 1691-1695.

[150] J. J. Filliben and A. Heckert, "Exploratory Data Analysis," in *NIST/SEMATECH e-Handbook of Statistical Methods*, J. J. Filliben, Ed., ed. Gaithersburg, MD: National Institute of Standards & Technology, 2011.

[151] M. Yuanbin, "Particle Swarm Optimization for Cylinder Helical GearMulti-objective Design problems," *Applied Mechanics and Materials*, vol. 109, pp. 216-221, 2012.

[152] M. Spasos, K. Tsiakmakis, N. Charalampidis, and R. Nilavalan, "RF-MEMS switch actuation pulse optimization using Taguchi's method," *Microsystem Technologies*, vol. 17, pp. 1351-1359, 2011.

[153] F. Ducatelle, G. A. D. Caro, and L. M. Gambardella, "Principles and applications of swarm intelligence for adaptive routing in telecommunications networks," *Swarm Intelligence*, vol. 4, pp. 173-198, 2010.

[154] M. Wan, L. Li, J. Xiao, C. Wang, and Y. Yang, "Data clustering using bacterial foraging optimization " *J. Intelligent Information Systems*, vol. 38(2), pp. 321-341, 2012.

[155] D. Karaboga and B. Basturk, "A Powerful and Efficient Algorithm for Numerical Function Optimization: Artificial Bee Colony (ABC) Algorithm," *J. Global Optimization*, vol. 39(3), pp. 459-171, 2007.

[156] I. Brajevic and M. Tuba, "An upgraded artificial bee colony (ABC) algorithm for constrained optimization problems," *J. Intelligent Manufacturing*, vol. 10 January 2012, pp. 1-12, 2012.

[157] P. Udhayakumar and S. Kumanan, "Sequencing and scheduling of job and tool in a flexible manufacturing system using ant colony optimization algorithm," *Int. J. Advanced Manufacturing Technology*, vol. 50, pp. 1075-1084, 2010.

[158] Y. Liu, Z. Zhang, and Z. Liu, "Customized configuration for hierarchical products: component clustering and optimization with PSO," *Int. J. Advanced Manufacturing Technology*, vol. 57, pp. 1223-1233, 2011.

[159] W.-A. Yang, Y. Guo, and W.-H. Liao, "Optimization of multi-pass face milling using a fuzzy particle swarm optimization algorithm," *Int. J. Advanced Manufacturing Technology*, vol. 54, pp. 45-57, 2011.

[160] D. I. Abu-Al-Nadi, T. H. Ismail, H. Al-Tous, and M. J. Mismar, "Design of Linear Phased Array for Interference Suppression Using Array Polynomial Method and Particle Swarm Optimization," *Wireless Personal Communications*, vol. 63(2), pp. 501-513, 2012.

[161] X.-L. Li, "A Particle Swarm Optimization and Immune Theory-Based Algorithm for Structure Learning of Bayesian Networks," *Int. J. Database Theory & Application*, vol. 3(2), pp. 61-70, 2010.

[162] M. M. K. El-Nagar, D. K. Ibrahim, and H. K. M. Youssef, "Optimal Harmonic Meter Placement Using Particle Swarm Optimization Technique," *Online J. Power and Energy Engineering*, vol. 2(2), pp. 204-209, 2011.

[163] T. Lan, Q. Kang, J. An, W. Yan, and L. Wang, "Sitting and sizing of aggregator controlled park for plug-in hybrid electric vehicle based on particle swarm optimization," *Neural Computing & Applications*, pp. 1-9.

[164] A.M.Moussa, M.E.Gammal, A.A.Ghazala, and A.I.Attia, "An Improved Particle Swarm Optimization Technique for Solving the Unit Commitment Problem," *Online J. Power and Energy Engineering*, vol. 2(3), pp. 217-222, 2011.

[165] M.-Y. Cheng, K.-Y. Huang, and H.-M. Chen, "Dynamic guiding particle swarm optimization with embedded chaotic search for solving multidimensional problems," *Optimization Letters*, vol. 6, pp. 719–729, 2012.

[166] X. Na, H. Dong, Z. GuoFu, J. JianGuo, and V. Khong, "Study on attitude determination based on discrete particle swarm optimization," *Science China Technological Sciences*, vol. 53(12), pp. 3397-3403, 2010.

[167] F.-B. Liu, "An application of particle swarm optimization to identify the temperature-dependent heat capacity," *Heat Mass Transfer*, vol. 48, pp. 99-107, 2012.

[168] R. K. Aggarwal and M. Dave, "Filterbank optimization for robust ASR using GA and PSO " *Int. J. of Speech Technology*, vol. 15(2), pp. 191-201, 2012.

[169] Z. M. Yusof, A. F. Z. Abidin, M. N. A. Salam, K. Khalil, J. A. A. Mukred, M. K. Hani, and Z. Ibrahim, "A Binary Particle Swarm Optimiazation Approach for Buffer Insertion in VLSI Routing," *Int. J. Innovative Management, Information & Production*, vol. 2(3), pp. 34-39, 2011.

[170] S. Karol and V. Mangat, "Use of Particle Swarm Optimization in Hybrid Intelligent Systems," *J. Information & Operations Management*, vol. 3(1), pp. 293-296, 2012.

[171] Y. M. B. Ali, "Psychological model of particle swarm optimization based multiple emotions," *Applied Intelligence*, vol. 36, pp. 649-663, 2012.

[172] A. E. Dor, M. Clerc, and P. Siarry, "A multi-swarm PSO using charged particles in a partitioned search space for continuous optimization," *Computational Optimization and Applications*, pp. 1-25, 2011.

[173] B. Cheng, S. Antani, R. J. Stanley, and G. R. Thoma, *Graphical image classification using a hybrid of*

evolutionary algorithm and binary particle swarm optimization. San Francisco, CA: submitted to SPIE Electronic Image, 2012.

[174] I. X. Tassopoulos and G. N. Beligiannis, "Using particle swarm optimization to solve effectively the school timetabling problem," *Soft Computing*, vol. 16, pp. 1229-1252, 2012.

[175] G. Folino and C. Mastorianni, "Special Issue: Bio-Inspired Optimization Techniques for High Performance Computing," *New Generation Computing*, vol. 29, pp. 125-128, 2011.

[176] R. C. Green, L. Wang, M. Alam, and R. A. Formato, "Central force optimization on a GPU: a case study in high performance metaheuristics," *J. Supercomputing*, pp. 1–21, 2012.

[177] M. R. Farmani, A. Jaamialahmadi, and M. Babaie, "Multiobjective optimization for force and moment balance of a four-bar linkage using evolutionary algorithms," *Journal of Mechanical Science & Technology*, vol. 25 (12), pp. 2971-2977, 2011.

[178] M. Vadood and D. Semnani, "Optimization of Fiber Distribution in Spunbond Non-woven Structure," *Fibers and Polymers*, vol. 12(6), pp. 821-829, 2011.

[179] A. Mishra and R. R. Sahay, "Well Parameters' Estimation using Traditional and Non-traditional Methods," *Int. J. Earth Sciences & Engineering*, vol. 4(6), pp. 235-238, 2011.

[180] A. A. Altun and M. A. Sahman, "Cost optimization of mixed feeds with the particle swarm optimization method," *Neural Computing & Applications*, pp. 1-8, 2011.